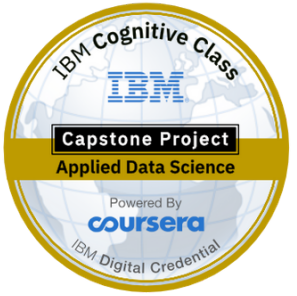
Restaurant location research in Dublin



IBM Data Science Capstone Project Report

Jose Hernandez

Dublin, June 2020

jh.hdez@gmail.com

Abstract

In this project, I investigate potential factors that may affect business performance when choosing the perfect location for a restaurant business in Dublin city. The goal is to check if a relationship exists between the proximity of a restaurant to diverse features, such as existing business and points of interest.. For the predictive model, I have used SVM, KNN and Decision Tree classifiers. I have used tools for feature selection, in order to generate a combination of results for all features, and to compensate for some skewed results.
I have used a 2047 combination of feature sets for each method within each classifier. I also included a method for Cross Validation modifying the random state parameter while splitting the dataset into train and test sets obtaining better results than the conventional K-Fold and cross_val_score methods.
As several factors affect a business success, the location being just one of them, this study should not be used on its own in order to make investment decisions.

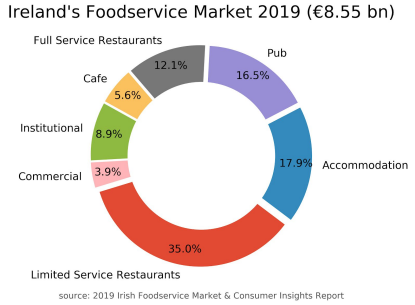
Table of contents

I.INTRODUCTION

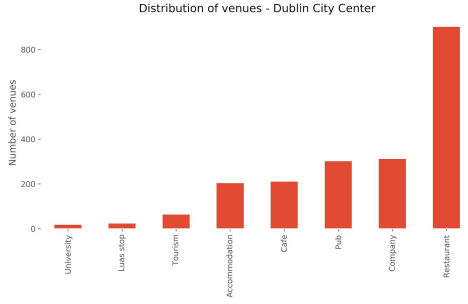
A. Background

Dublin is the capital of Ireland and located in the province of Leinster on the country's eastern coast and spread out over 318 square kilometres. Dublin has 1 million people living in the city and 1.2 million living in the contiguous urban area. Dublin has one of the highest ratios of tourists to locals in the world, surpassing New York, London and Paris. Dublin attracted over 11 million visitors in 2019, about 427 tourists for every 100 hundred locals in the capital.

A market share analysis of the food services industry in Dublin Bord Bia 2019 Irish Foodservice Market & Consumer Insights Report found that 8.5 billion Euro was spent by visitors to the region on food services in 2019.



Foodservice or 'Out of Home' is the term used to describe all food consumed and prepared out of home. Almost half of the market (47.1%) in Dublin is shared by Full Service Restaurants, and Limited Service Restaurants, like fast food businesses. We can see the distribution of venues in Dublin City, collected using Foursquare API.



B. Problem

Customers visit a restaurant for the atmosphere and come back for the food and service.

A large part of a restaurant's success is its reputation which is the main reason why people continually visit. So can we say that the above statement applies to visitors to a region who are only there for a short period of time?

As with all businesses, there are many factors which influence the success or otherwise of a restaurant - food quality, service, experienced professionals and value for money, amongst others. But what part does the location of a restaurant play in its success?

There are some aspects to take into account when choosing the location:

- *Population* in Dublin is growing, albeit also with the number of new businesses.
- *Accessibility* is a key factor, so being situated close to a tram station would be a smart idea.
- *Surrounding businesses*, such as cafes or pubs, will entice people to these areas who initially may be just looking for a coffee and/or sociable drink. Locating in areas with a high footfall of similar outlets will highlight a location as a place to go for something to eat and drink. After all, we still are social individuals.
- *Affordability* will be higher in some districts far from the core of the city, however, will also have a lack of *visibility*, which is higher where tourists go, impacting on the business success.
- *Crime Rates* have a homogeneous distribution within the city center. Although there are some niches where crime rate is noticeable, the data from the central police station covers all of Dublin City, so for this project will not be an essential feature to consider. Moreover, in this case, crime rates are correlated to the districts, so having the price area by district will have implicit this data on the study.

C. Interest

Due to Covid-19 there is uncertainty in the market at the time of carrying out this study. However it is considered that there is a long term investment opportunity for stakeholders who have the resources to conduct a full scale cost benefit analysis.

Getting into the scope of this study, we can consider some questions, like if there is any relationship between restaurant success and the closeness to:

- ❏ the river Liffey that crosses the city
- ❏ a Luas tram station helping accessibility to your business
- ❏ accommodations, such as Bed & Breakfast, Hostels or Hotels
- ❏ tourist attractions and museums
- ❏ cafes
- ❏ pubs
- ❏ companies
- ❏ universities
- ❏ other restaurants

Is there any combination of the mentioned factors that would point out where stakeholders should focus on? Well, as Data Science can offer a new perspective in the matter, we will go through all the different features set using several machine learning tools in order to predict which would be the optimal solution.
We can have a closer look at these factors and the reason why they have been included into the study enumerating some facts:

Luas (Dublin City tram)
On average 500,000 people travel within Dublin City Centre every day. This is made up of circa 235,000 work-related trips, 45,000 education trips, and 120,000 visitors, tourists, and shoppers. 42 million passenger journeys were made using the Luas in 2018. This is 23% of daily commuting.

Accommodation
Dublin had the highest hotel occupancy rate among European cities in 2018 (83.8%). Over 200 places in Dublin city offer accommodation services.

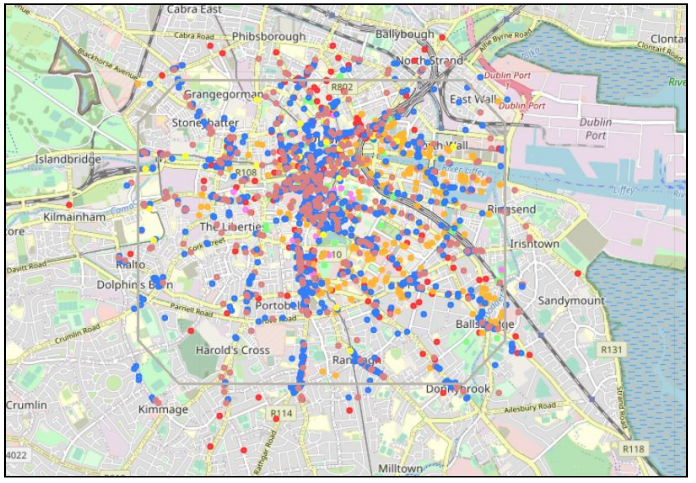
Tourism
Most popular tourist attractions in Ireland are in Dublin. Collectively, the Guinness Storehouse and the Book of Kells attracted over 2.7 million tourists in 2019.

Cafe
An European style coffee culture in Dublin has risen in popularity over the past number of years. There are over 200 coffee shops in Dublin City center confirming the proliferation of cafes with specialist baristas due to rising demand.

Pub
Irish pubs are the core of the city. With over 300 pubs in the city center we can see most of them are located in the famous Temple Bar area, mixed with many restaurants. Pubs in tourist-oriented areas like Dublin City are also more likely to serve food to their customers.

Companies
Over 100,000 people are employed in Dublin in sectors such as Pharmaceuticals, Medical & Dental Supplies, Tech companies and Engineering.
Dublin has the top 5 global software companies, 9 of the world's top 10 pharmaceutical companies, half of the world's top 50 banks, 250 global financial institutions, 12 of the world's top 20 insurance companies and 18 of the world's top 25 med tech companies located in the city and its environs.

Universities
Approximately 120,000 students in total attend the Dublin region's five universities. The oldest, Trinity, is in the heart of the city.
In the next map we can notice the distribution of venues in Dublin City, concentrated in the core of the city.



Tourism, Restaurant, Accommodation, University, Cafe, Pub, Companies, Luas stop, Project boundary

II. DATA

A. Project Scope Boundaries

Firstly we will create a mesh of points on top of Dublin city due to asymmetrical boroughs shape, so we can use each point in order to explore the area leveraging the Foursquare API. Although we will get extra duplicates, these are easy to handle using pandas on python.

The advantage is the lack of missing venues if we were using the API in the conventional way. Shown in the picture, the mesh with 76 points and coordinates.

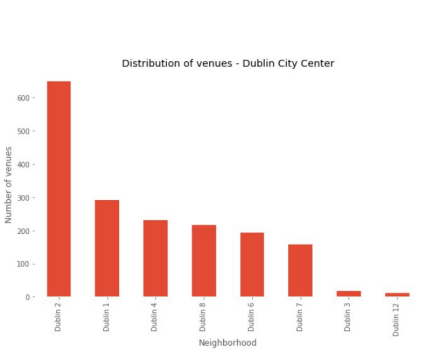
Dataframe with district and postcode information for each single one.

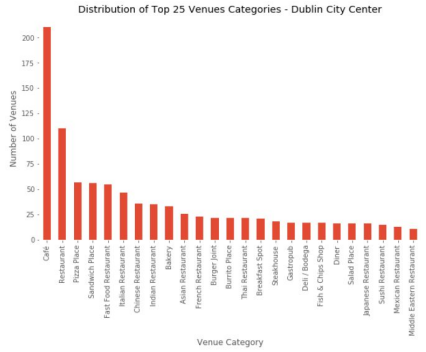
	Latitude	Longitude	Coordinates	District	PostCode
0	53.322556	-6.287982	53.322556, -6.287982	Kimmage C ED	Dublin 6
1	53.322556	-6.280417	53.322556, -6.280417	Kimmage C ED	Dublin 6
2	53.322556	-6.272853	53.322556, -6.272853	Rathmines West F ED	Dublin 6
3	53.322556	-6.265288	53.322556, -6.265288	Rathmines and Rathgar West ED	Dublin 6
4	53.322556	-6.257723	53.322556, -6.257723	Rathmines and Rathgar West ED	Dublin 6

B. Foursquare API

First I used the explore API endpoint in order to get venues around each point of the mesh. After that, I repeated it with a specific category parameter, like food, hotel, etc.

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Dublin 12	53.322556	-6.291964	Pickles Deli	53.321021	-6.293015	Deli / Bodega
1	Dublin 12	53.322556	-6.291964	Monto Cafe	53.320752	-6.292583	Diner
2	Dublin 12	53.322556	-6.291964	Apache Pizza	53.322971	-6.295267	Pizza Place
3	Dublin 12	53.322556	-6.291964	Matt The Rashers	53.320459	-6.291814	Café
4	Dublin 6	53.322556	-6.281214	Craft Restaurant	53.322698	-6.279316	Restaurant





C. Google Places API

Although Foursquare gave us quite a bit of information, due to Foursquare limitations and also because in a Data Science project like this the data to work with is crucial in order to get satisfactory results, I used Google API extending the datasets we already had obtained.

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue id	Venue price level	Venue Rating	Total user ratings	Venue Category
Dublin 3	53.356207	-6.242594	Dublin	53.349805	-6.260310	71ccb3b82cc1f534c1da9998f8610d85a6d7c598	0	0.0	0	[locality, political]
Dublin 3	53.356207	-6.242594	Jurys Inn Christchurch Dublin	53.342832	-6.270520	2ccbdecd28694cc8b74bfe56279db7fb52187101	0	4.2	1320	[lodging, restaurant, food, point_of_interest,...
Dublin 3	53.356207	-6.242594	Kinlay House Dublin	53.343726	-6.269898	7511e298b05c19078d13d98bb64909a1e62549f7	0	3.9	1110	[lodging, point_of_interest, establishment]
Dublin 3	53.356207	-6.242594	Radisson Blu Royal Hotel, Dublin	53.340853	-6.268325	5245b21949a3fad5785f03ff1427446a7339d412	0	4.3	1363	[lodging, point_of_interest, establishment]

D. Web Scraping

1. Restaurant Ranking

First we get the list of the ranking of all restaurants in Dublin City from TA website.

Name	Ranking	Link
Tang Cafe	1	/Restaurant_Review-g186605-d10387074-Reviews-T...
Taza	2	/Restaurant_Review-g186605-d17181958-Reviews-T...
Glovers Alley	3	/Restaurant_Review-g186605-d13477650-Reviews-G...

Once the list is retrieved I wrote a different script for web scraping the details of each single restaurant creating a dataframe with all the information. We also use Geopy libraries in order to get its coordinates. For the restaurant that we could not get this information we will use Google API.

	Name	Ranking	Reviews	Rating	Price	Cuisines	Address	Phone	Link	Geopy Latitude	Geopy Longitude
0	Tang Cafe	1	502	5.0	Cheap	['Cafe', 'European', 'Healthy']	23c Dawson Street, Dublin D02 PW18 Ireland	+353 86 391 5401	/Restaurant_Review-g186605-d10387074-Reviews-T...	53.348720	-6.258399
1	Taza	2	178	5.0	Average	['Indian', 'Asian', 'Pakistani']	2 Ardcollum Avenue Artane, Dublin D05 XW88 Ire...	+353 1 558 2866	/Restaurant_Review-g186605-d17181958-Reviews-T...	NaN	NaN
2	Glovers Alley	3	193	5.0	Expensive	['Irish', 'European', 'Vegetarian Friendly']	128 Stephen's Green Fitzwilliam Hotel, Dublin ...	+353 1 244 0733	/Restaurant_Review-g186605-d13477650-Reviews-G...	53.339644	-6.263466

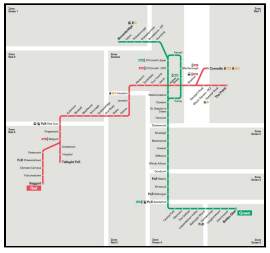
After that, we just remove restaurants outside the project scope boundaries based on latitude and longitude coordinates.

We'll compare the correlation between Rating and Ranking in order to filter some samples that have outliers, like high Rating but low Ranking due to inconsistent data from the website from where we had collected the data . After that, we'll define Rating as the KPI to measure the performance of a business, therefore, as the target with the predictive models we'll use in the project.

After cleaning the restaurant's dataset, we obtained 1185 samples.

2. Luas stops (city tram)

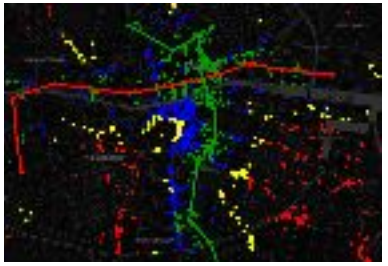
Data has been collected by web scraping the official website.



	Name	Address	Latitude	Longitude
0	Tallaght Luas stop	Oldbawn, Dublin, Ireland	53.287367	-6.374592
1	Saggart Luas stop	Saggart Luas Stop, Fortunestown, Saggart, Co. ...	53.284641	-6.437762
2	Fortunestown Luas stop	Fortunestown, Tallaght, Dublin, Ireland	53.284210	-6.424610
3	Citywest Campus Luas stop	Citywest Campus Luas Stop, Cooldown Commons, D...	53.287800	-6.418820
4	Cheeverstown Luas stop	Cheeverstown Luas Stop, Tallaght, Dublin, Ireland	53.291036	-6.406877

Once data had been collected, we calculated the distance between each restaurant to the closest Luas stop, fixing the minimum distance as 100 meter. This means, a restaurant will get the same weight value once it has a Luas stop in a range of 100 meters.

Statistics of all our samples versus Luas and closest restaurants in green shown in the map.



	Latitude	Longitude	Distances min
count	1188.000000	1188.000000	1188.000000
mean	53.341764	-6.260744	396.907407
std	0.007607	0.012098	347.031231
min	53.322580	-6.295349	100.000000
25%	53.337302	-6.265706	155.000000
50%	53.343006	-6.262388	292.500000
75%	53.347322	-6.255040	476.500000
max	53.356205	-6.227568	1834.000000

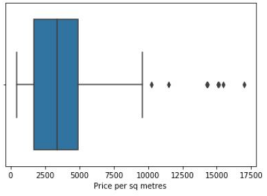
3. Price Area

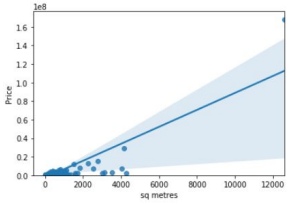
By web scraping the leader website Daft.ie where Estate Agents publish their ads we collected all the ads for commercials in Dublin.

	Name	Commercial Type	Price	sq metres	Neighbourhood
0	11/11A Ormond Quay Lower, Dublin 1	Office To Let or For Sale	1300000	465	Dublin 1
1	21 Ormond Quay Upper, Dublin 1	Restaurant / Bar / Hotel For Sale	2250000	570	Dublin 1
2	32 Lower Ormond Quay, Dublin 1, Dublin 1	Investment Property For Sale	700000	314	Dublin 1
3	308 The Capel Building, Mary Street, Dublin 1	Office For Sale	180000	NaN	Dublin 1
4	Independent House, Talbot Street, Dublin 1	Investment Property For Sale	29000000	NaN	Dublin 1

	Price	sq metres	Price per sq metres
count	1.730000e+02	173.000000	173.000000
mean	1.373641e+06	585.901734	5624.466243
std	2.076866e+06	1206.434887	26487.273172
min	1.000000e+00	4.000000	0.010000
25%	3.950000e+05	154.000000	1325.300000
50%	6.500000e+05	254.000000	3232.760000
75%	1.400000e+06	546.000000	4828.800000
max	1.500000e+07	13152.000000	350000.000000

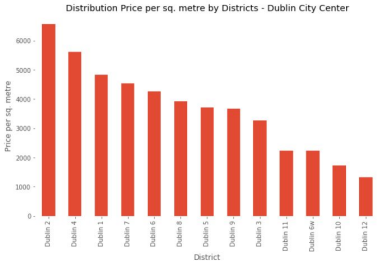
Data had to be cleaned and corrected due to some advertisements had no price, or some had wrong information published on the website.
After visualizing the data we could identify some outliers and remove them from the dataframe.





	Name	Commercial Type	Price	sq metres	Neighbourhood	Price per sq metres
0	Cleary's Bar, Lounge & Adjacent Barbers Shop, ...	Restaurant / Bar / Hotel For Sale	1500000.0	301.0	Dublin 1	4983.39
1	21 Ormond Quay Upper, Dublin 1	Restaurant / Bar / Hotel For Sale	2250000.0	570.0	Dublin 1	3947.37

Chart with the means for each district and visualization on a heatmap.



E. Liffey (river)

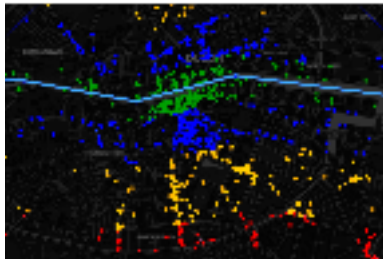
After choosing 5 points which describe the Liffey trajectory, we divided each segment using python in order to get extra points to calculate distances between each restaurant and the closest part of the river.

Once the data is calculated, we normalize it and save it for later use. Similar to the previous feature, we fixed a minimum distance of 200 meters.

	Name	Ranking	Reviews	Rating	Price	Cuisines	Address	Latitude	Longitude	Phone	Link	Distance	Distance Proximity
59	Umi Falafel	73	1008	4.5	Cheap	['Lebanese', 'Fast food', 'Mediterranean']	Callaghan House 13 Dame Street, Dublin D02HX67...	53.324603	-6.265240	+353 1 670 6866	/Restaurant_Review-g186605-d5003310-Reviews-Um...	2344.58	Farest
229	The Brewer's Dining Hall - Guinness Storehouse	267	494	4.5	Average	['Irish', 'European', 'Vegetarian Friendly']	St. James Gate Saint James's Gate, Dublin Dubl...	53.344112	-6.285172	+353 1 408 4800	/Restaurant_Review-g186605-d1554254-Reviews-Th...	284.904	Farest

Statistics of all our samples versus Liffey and closest restaurants in green shown in the map.

	Ranking	Reviews	Rating	Latitude	Longitude	Distance
count	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000
mean	867.664141	277.626263	4.141835	53.341764	-6.260744	762.503896
std	583.001783	555.012944	0.568664	0.007607	0.012098	648.587962
min	1.000000	1.000000	1.000000	53.322580	-6.295349	200.000000
25%	353.750000	12.000000	4.000000	53.337302	-6.265706	252.670755
50%	782.500000	71.000000	4.000000	53.343006	-6.262388	514.833512
75%	1353.750000	299.250000	4.500000	53.347322	-6.255040	1057.797971
max	2045.000000	6164.000000	5.000000	53.356205	-6.227568	2741.908314



F. Companies

A csv with a list of companies was obtained from IDA Ireland.

	Company	Sector(s)	Product	Nationality	Address
0	Symantec Limited	ICT- ICT - Software	Package Software for PC and Mac and technical ...	United States of America	Ballycoolin Industrial Park Blanchardstown Du...
1	Ig International Management Limited	Financial Services	Asset Management - owned by Power Financial Gr...	Canada	Brooklawn House Shelbourne Road Ballsbridge Du...

After filtering the data for businesses in Dublin, we used Geopy to get the coordinates, and alternatively Google API for those company names or addresses that Geopy could not resolve.

	Company	Address	Latitude	Longitude
0	Symantec Limited	Ballycoolen, Dublin 15, Ireland	53.412959	-6.373212
1	Ig International Management Limited	Brooklawn House, Shelbourne Rd, Ballsbridge, D...	53.332099	-6.230495

G. Districts Geoshapes

Dublin Postcodes extracted from a GeoCluster repository and converted to shapefile in order to import it with geopandas.

```
   id                                           geometry
0   0  POLYGON ((-6.33272 53.37071, -6.33274 53.37072...
1   1  POLYGON ((-6.25925 53.36170, -6.24931 53.35908...
2   2  POLYGON ((-6.26780 53.34588, -6.25454 53.34802...
3   3  POLYGON ((-6.21911 53.35739, -6.22102 53.35491...
4   4  POLYGON ((-6.23737 53.34359, -6.23052 53.34304...
```

	Link	Latitude	Longitude	geometry
0	/Restaurant_Review-g186605-d10387074-Reviews-T...	53.34872	-6.258399	POINT (-6.25840 53.34872)

	Link	District
0	/Restaurant_Review-g186605-d10387074-Reviews-T...	1.0
1	/Restaurant_Review-g186605-d13477650-Reviews-G...	2.0
2	/Restaurant_Review-g186605-d6403998-Reviews-Da...	8.0
3	/Restaurant_Review-g186605-d2239110-Reviews-Mu...	4.0
4	/Restaurant_Review-g186605-d15590976-Reviews-T...	8.0

Getting the list of restaurants and transforming the dataframe in a geodataframe

After we just merge the geodataframes to get district data into our main dataframe. Due to many samples had missing district information, or were wrong, we used the coordinates to tackle this issue.

	Name	Ranking	Reviews	Rating	Price	Cuisines	Address	Latitude	Longitude	Phone	Link	District
0	Tang Cafe	1	502	5.0	Cheap	['Cafe', 'European', 'Healthy']	23c Dawson Street, Dublin D02 PW18 Ireland	53.348720	-6.258399	+353 86 391 5401	/Restaurant_Review-g186605-d10387074-Reviews-T...	1.0
1	Glovers Alley	3	193	5.0	Expensive	['Irish', 'European', 'Vegetarian Friendly']	128 Stephen's Green Fitzwilliam Hotel, Dublin ...	53.339644	-6.263466	+353 1 244 0733	/Restaurant_Review-g186605-d13477650-Reviews-G...	2.0
2	Darkey Kelly's Bar & Restaurant	4	1645	4.5	Average	['Irish', 'Bar', 'European']	Fishamble Street ChristChruch, Dublin Ireland	53.343513	-6.271060	+353 83 346 4682	/Restaurant_Review-g186605-d6403998-Reviews-Da...	8.0
3	Mulberry Garden	5	909	4.5	Expensive	['Irish', 'European', 'Contemporary']	Mulberry Lane Donnybrook, Dublin 04 Ireland	53.322659	-6.236801	+353 1 269 3300	/Restaurant_Review-g186605-d2239110-Reviews-Mu...	4.0
4	The Landmark	6	468	4.5	Average	['Irish', 'Bar', 'European']	The Landmark 40 Wexford Street, Dublin D02 CH6...	53.337441	-6.265903	+353 1 537 9951	/Restaurant_Review-g186605-d15590976-Reviews-T...	8.0

H. Feature Generation

We are taking into account 4 different ranges for features such as **Pubs**, **Tourism**, **Cafes**, **Universities**, **Accommodations**, **Companies**, **Museums** and **Restaurants**. We calculated the amount of each category within that range versus every single restaurant in our sample data.

- < 250 meters
- 250 to 500 meters
- 500 to 1000 meters
- 1000 to 2000 meters

Once calculated the number of venues per range we multiply each one for a fixed number, which is proportional to the proximity to the restaurant. For instance, within the range of 250 meters, it would be 2000/250, and so on.

Sample of a restaurant with its weights calculated:

	Link	Hotel weight	Tourism weight	University weight	Cafe weight	Pub weight	Museum weight	Company weight	Restaurant weight
	/Restaurant_Review-g186605-d953880-Reviews-The...	381.0	120.0	35.0	394.0	1028.0	93.0	353.0	2711.0

Also the describe method giving us detailed information about the dataframe:

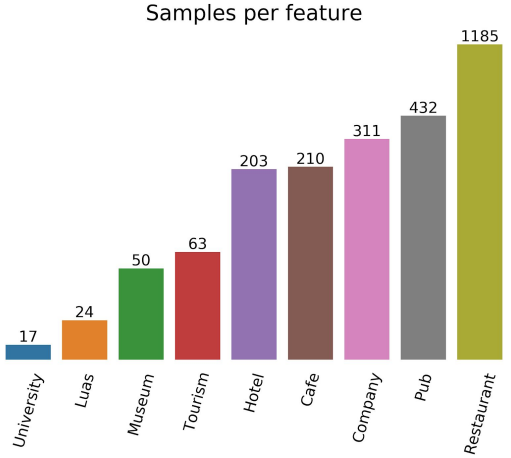
	Ranking	Reviews	Rating	Latitude	Longitude	Hotel weight	Tourism weight	University weight	Cafe weight	Pub weight	Museum weight
count	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000
mean	867.664141	277.626263	4.141835	53.341764	-6.260744	258.089226	89.207071	27.739899	277.725589	652.114478	67.496633
std	583.001783	555.012944	0.568664	0.007607	0.012098	101.944971	39.773485	13.983280	103.940762	302.811160	27.388912
min	1.000000	1.000000	1.000000	53.322580	-6.295349	11.000000	0.000000	0.000000	20.000000	21.000000	1.000000
25%	353.750000	12.000000	4.000000	53.337302	-6.265706	181.000000	60.000000	18.000000	197.000000	405.000000	47.000000
50%	782.500000	71.000000	4.000000	53.343006	-6.262388	274.500000	93.000000	27.000000	292.500000	646.500000	72.000000
75%	1353.750000	299.250000	4.500000	53.347322	-6.255040	345.250000	125.000000	39.000000	383.000000	904.000000	93.000000
max	2045.000000	6164.000000	5.000000	53.356205	-6.227568	410.000000	148.000000	66.000000	424.000000	1204.000000	111.000000

After that, we normalized the dataframe using scikit learn library *preprocessing.StandardScaler* in order to use them later.

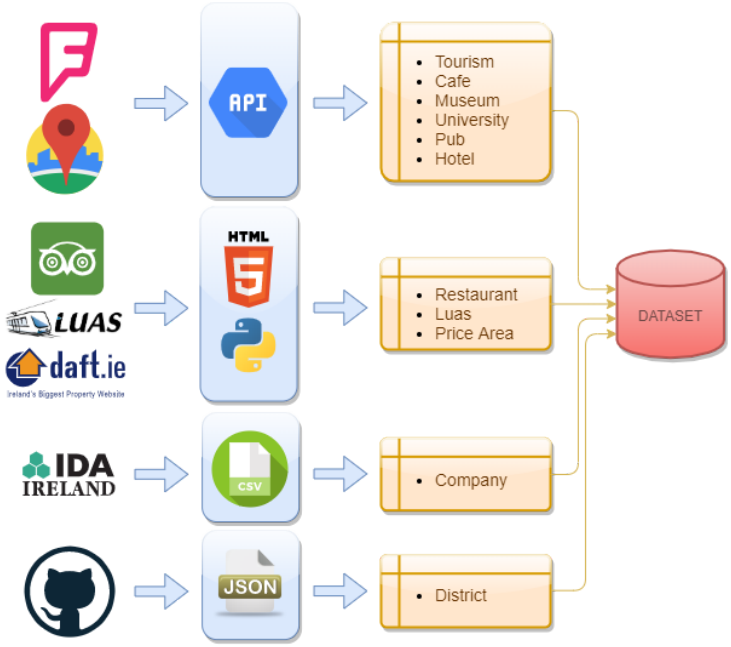
	Ranking	Reviews	Rating	Latitude	Longitude	Hotel weight NOR	Tourism weight NOR	University weight NOR	Cafe weight NOR	Pub weight NOR	Museum weight NOR
count	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000	1188.000000
mean	867.664141	277.626263	4.141835	53.341764	-6.260744	2.532719	2.243822	1.984626	2.673086	2.154442	2.465416
std	583.001783	555.012944	0.568664	0.007607	0.012098	1.000421	1.000421	1.000421	1.000421	1.000421	1.000421
min	1.000000	1.000000	1.000000	53.322580	-6.295349	0.107947	0.000000	0.000000	0.192498	0.069379	0.036527
25%	353.750000	12.000000	4.000000	53.337302	-6.265706	1.776215	1.509178	1.287794	1.896109	1.338030	1.716746
50%	782.500000	71.000000	4.000000	53.343006	-6.262388	2.693763	2.339226	1.931691	2.815288	2.135893	2.629908
75%	1353.750000	299.250000	4.500000	53.347322	-6.255040	3.388057	3.144121	2.790220	3.686343	2.986616	3.396965
max	2045.000000	6164.000000	5.000000	53.356205	-6.227568	4.023471	3.722639	4.721910	4.080965	3.977750	4.054442

I. Feature Summary

Once data has been cleaned and grouped into a dataset we obtained the amount per feature shown in the chart.

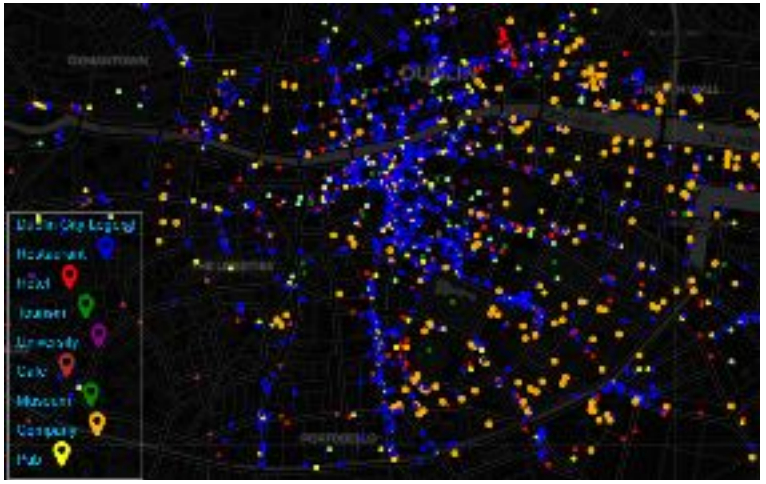


Flow diagram showing the data acquisition and processes before having the dataset ready to create the predictive models.



J. Interactive map

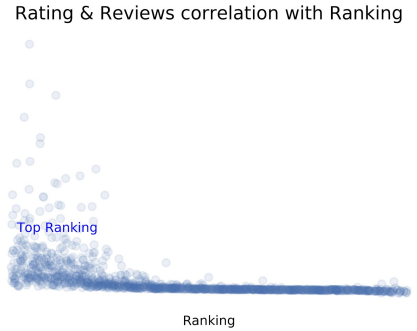
Next map with interactive layers can be found within this project's repository.



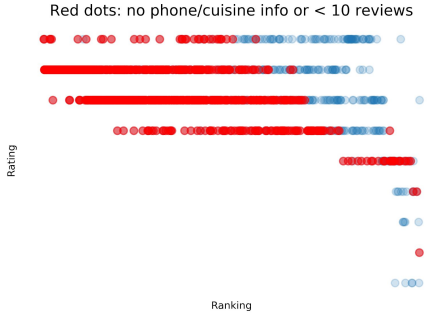
III. METHODOLOGY

A. Exploratory Data Analysis

First approach was to use Rating & Reviews as potential targets, however after analyzing the dataset we noticed there was an unbalanced distribution of Reviews. These were mostly within the top ranking list, which makes sense, albeit it is not useful for the project.

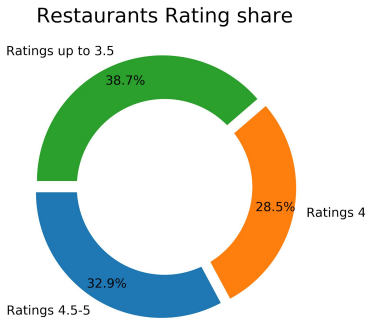


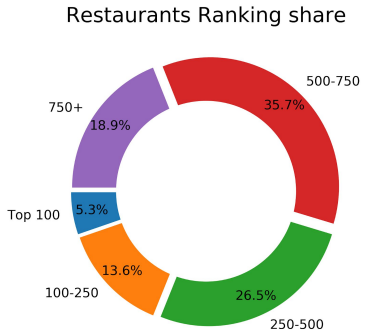
In order to filter some restaurant samples we need to remove unnecessary data. As shown in the chart, red dots would be restaurants with no phone or cuisine information or less than 10 reviews.



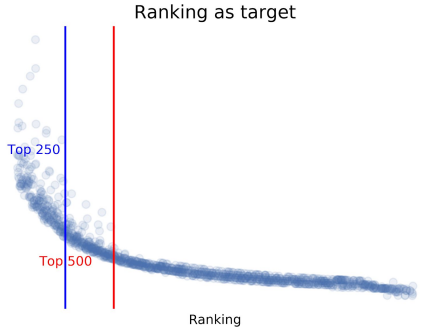
As we can tell, it makes no sense to see these restaurants with 3.5+ Rating. We can assume that this type of restaurant shouldn't be mixed on the Range of restaurants with 3.5+ Rating. This is due to a misclassification by the website from which we had collected the data. There are 246 restaurants without phone or cuisine information or less than 10 reviews and higher Rating than 3.5.

After cleaning the data, we obtained the next rating and ranking distributions in the dataset to work with.

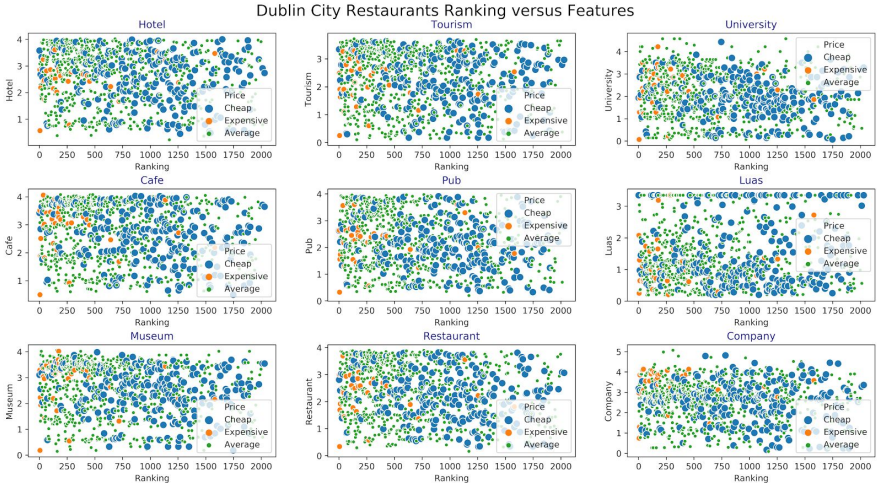


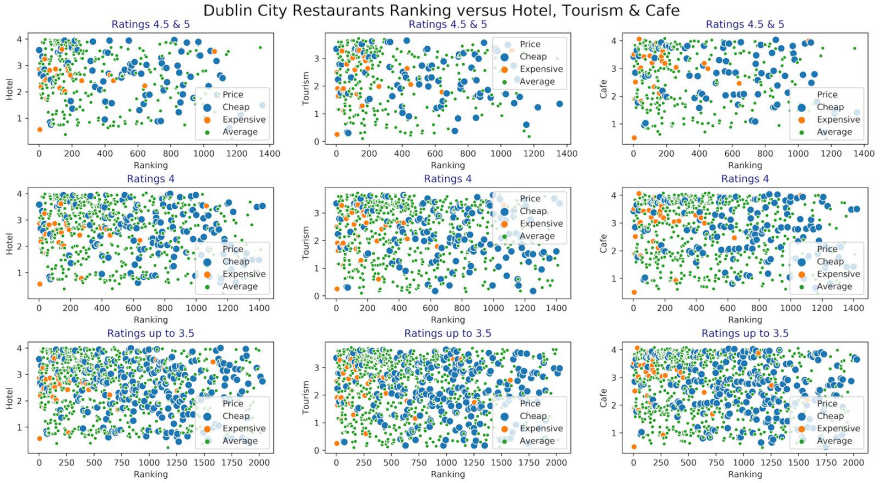


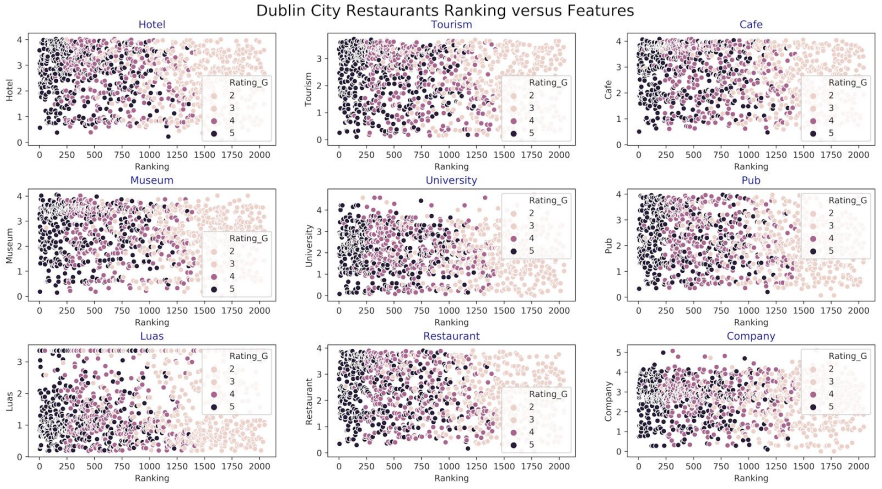
The idea is to have a predictive model where we can foretell if a new location for a restaurant will share similarities with current Top 250 & Top 500 ranges. Therefore, we will be able to use Data Science in order to add a probability of business success related to the location. This variable combined with other factors will provide an edge for a potential investment.

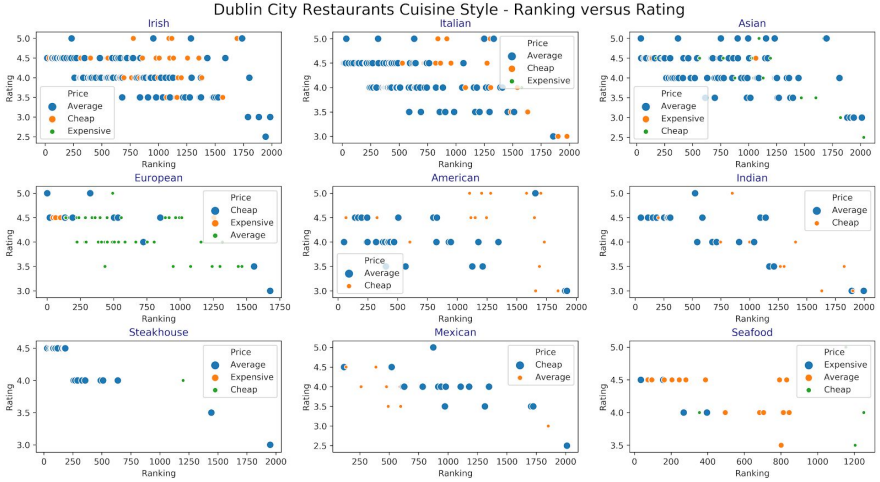


On the next page there are several charts showing ranking versus different features and the distribution on rating, average price of the restaurant and cuisine's style. We noticed a concentration of dots on the left upper corner, specially on charts with features as Hotel, Cafe and Tourism.


Dublin City Restaurants Ranking versus Features
Hotel
Tourism
University
Cafe
Pub
Luas
Museum
Restaurant
Company
Ranking
Price
Cheap
Expensive
Average


Dublin City Restaurants Ranking versus Hotel, Tourism & Cafe
Ratings 4.5 & 5
Ratings 4
Ratings up to 3.5
Hotel
Tourism
Cafe
Ranking
Price
Cheap
Expensive
Average


Dublin City Restaurants Ranking versus Features
Hotel
Tourism
Cafe
Museum
University
Pub
Luas
Restaurant
Company
Ranking
Rating_G
2
3
4
5


Dublin City Restaurants Cuisine Style - Ranking versus Rating
Irish
Italian
Asian
European
American
Indian
Steakhouse
Mexican
Seafood
Rating
Ranking
Price
Average
Cheap
Expensive

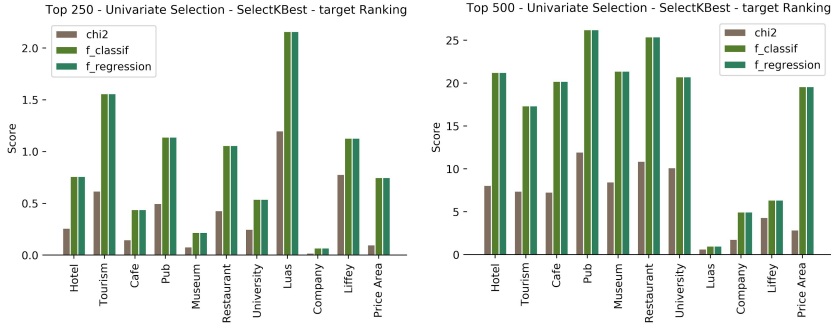
B. Inferential Statistics

When Y is discrete, we refer to the model as a classification model. Next we are having several approaches in order to improve the *Classification Feature Selection.*

1. Univariate Selection - SelectKBest

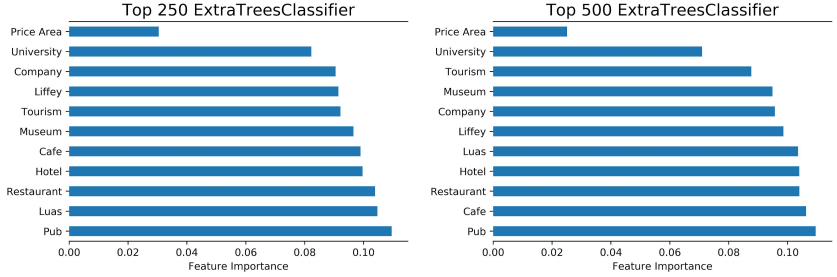
Three methods for classification tasks using targets Top_250 and Target_500:

- chi2: Chi-squared stats of non-negative features
- f_classif: ANOVA F-value between label/feature
- f_regression: F-value between label/feature



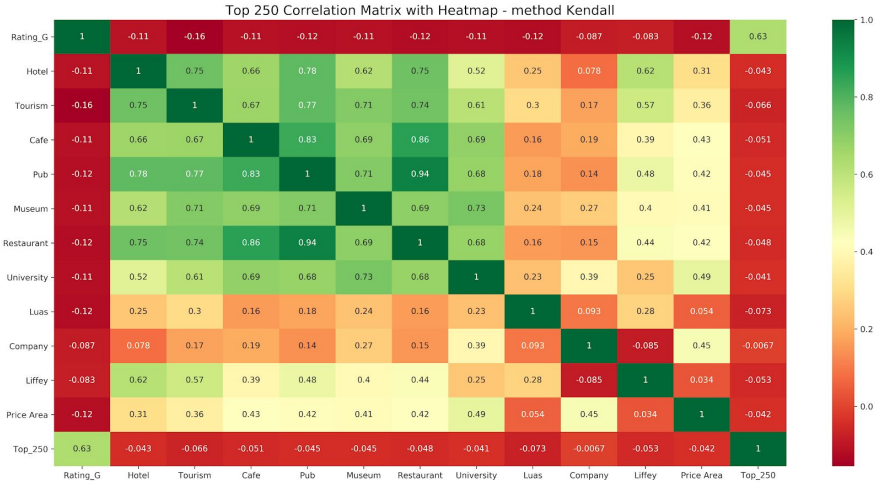
The differences between Top_250 and Top_500 will be detailed in the next chapter *Prediction Models*. In a nutshell, all the classifiers were trained for two targets in parallel, so from a practical standpoint we are obtaining two points of view of how the classifiers respond to a different amount of data given.

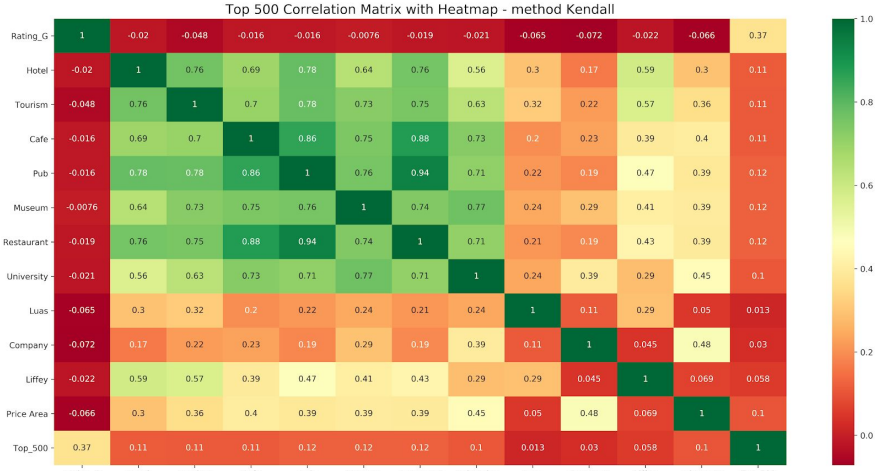
2. Feature Importance - ExtraTreesClassifier



3. Correlation Matrix with Heatmap - method Kendall

For uncorrelated features, the optimal feature size is N-1 (N sample size)
As feature correlation increases, the optimal feature size is proportional to √(N) for highly correlated features.
However, a rule of thumb for a dataset of N samples would be N/10 features





4. Scikit Feature Selection Algorithms

Several algorithms were used and grouped results into a table for comparison purposes..

- Pearson's correlation coefficient (linear)
- Recursive Feature Elimination (REF)
- Lasso: SelectFromModel - Logistic Regression
- Tree-based: SelectFromModel - Random Forest Classifier
- LightGBM Classifier

Top_250

	Feature	Pearson	RFE	Logistics	Random Forest	LightGBM	Total
1	Restaurant	True	True	True	True	False	4
2	Luas	True	True	False	True	True	4
3	Tourism	True	True	True	False	False	3
4	Pub	True	False	False	True	True	3
5	Liffey	True	False	False	True	True	3
6	Hotel	True	True	False	True	False	3
7	Cafe	False	True	True	True	False	3
8	Museum	False	True	True	False	False	2
9	University	False	False	False	False	True	1
10	Company	False	False	False	False	True	1
11	Price Area	False	False	False	False	False	0

Top_500

	Feature	Pearson	RFE	Logistics	Random Forest	LightGBM	Total
1	Pub	True	True	True	True	True	5
2	Hotel	True	True	True	True	True	5
3	Cafe	True	True	True	True	True	5
4	Restaurant	True	True	True	True	False	4
5	Museum	True	True	True	False	False	3
6	Luas	False	False	False	True	True	2
7	Liffey	False	False	False	True	True	2
8	University	True	False	False	False	False	1
9	Tourism	False	True	False	False	False	1
10	Company	False	False	False	False	True	1
11	Price Area	False	False	False	False	False	0

C. Prediction Models

Target generation:

Once we have the dataset ready to work, we can have a closer look at the ranking distribution.

samples	*ranking*
212	<= 250
412	<= 500
811	<= 1500
40	> 1500

As we see, these numbers are around the double from the previous one below 1500 ranking. So for each target we will work with a different number of samples limiting the data's scope in order to keep y-variance close to the unit. Doing this, the classifier will work under an unbiased training achieving a balanced model to use with classifiers. We also define target values of (1,-1) for SVM/DTree methods, and (1,0) for KKN.

We added two columns to the dataset filtering by ranking:

- <= 500 for Top_250
- <= 1500 for Top_500

The 40 samples above 1500 ranking will be used for the validation set. Therefore, all validation tests should return as negative in order to achieve 100% accuracy.

Instead of trying a feature set that has been chosen during Feature selection, we will calculate all possible combinations of feature sets. With each one, we run the classifier and get the accuracy. Albeit we had estimated the best features with different tools during the Feature Selection process, with a dataset of about 1000 samples this is an interesting approach due to these calculations won't need too much computing resources.

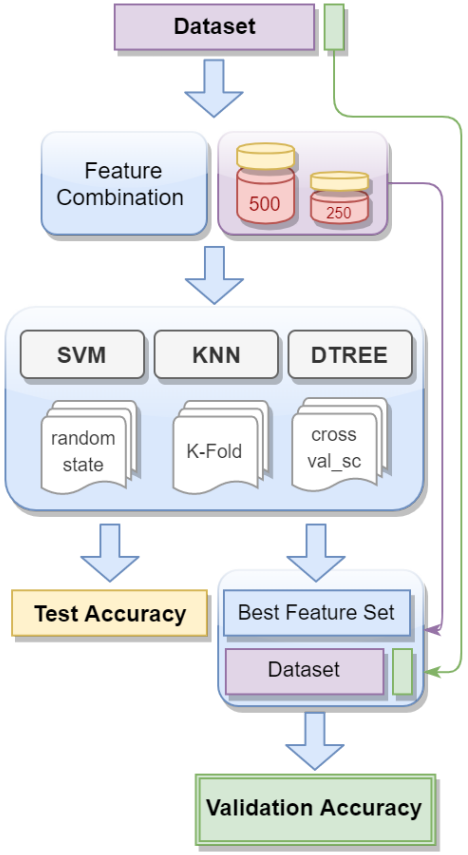
Out of the 11 features we generated 2047 different combinations of feature sets.

Methodology:

First we split the full database into a database for the classifiers and the validation set. After this, we generate the different combinations of features which will be the set of independent variables. We train the classifiers for the Top_250 and Top_500 (dependent variables) using several cross validation methods:

- Modifying the random_state parameter while we split the data into train and test sets
- K-Fold method with 10 folds
- Cross_val_score from scikit with 10 folds

At this point we have obtained the train and test accuracies. With this, we pick the best feature sets and the accuracies they got. We generate the ROC Curve in order to check the true and false positive rates.



Finally we use the validation set running the classifiers again with the best feature set, obtaining the validation accuracy for Top_250 and Top_500.
Detailed steps will be shown for each classifier.

1. Support Vector Machine Classifier

SVM works very well on smaller data sets, on non-linear data sets and high dimensional spaces. Also it has the advantage that it is not very sensitive to overfitting.
When the data set has more noise (i.e. target classes are overlapping) SVM doesn't perform well. We need to find classes that do not overlap. This is why using ranking as the target we obtain better results than using Rating.

- We are running the SVM classifier for all combinations of features.
- random_state from 0 to 100 and multiple of 3. This will serve as Cross Validation. Otherwise, fixing the seed (random_state) which generates the test set, wouldn't get optimal results.
- Data has been splitted in 80/20 ratio, 80% for train set and 20% for test set.
- Regularization Parameter C has been fixed to 5 due to previous tests with this model. There's no increment on accuracy with values above 10.
- Gamma parameter gave us better results slightly modifying the default (from version SVM 0.22) 'scale', which is 1/(n_features * X.var()). We are using (1.15 / (X.shape[1] * X.var()))
- Kernels ('linear', 'poly', 'rbf', 'sigmoid') also have been tested. Radial Basis Function was the most accurate followed by Polynomial. This is due to our model being non-linear.
- F1-measure, which weights precision and recall equally, is the variant most often used when learning from imbalanced data. We cannot base our study just on accuracy for this model in particular.

a) SVM Cross Validation iterating random state parameter

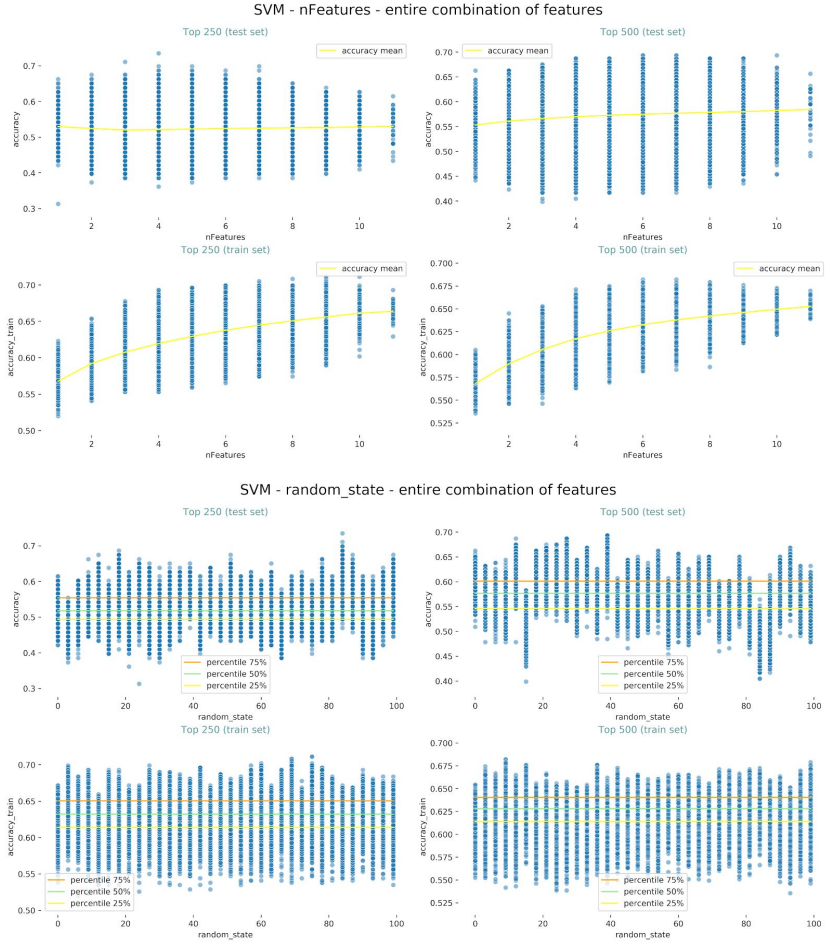
Sorting results we can see the max test accuracy of the model and the train accuracy for that feature set and parameters.

Top_250

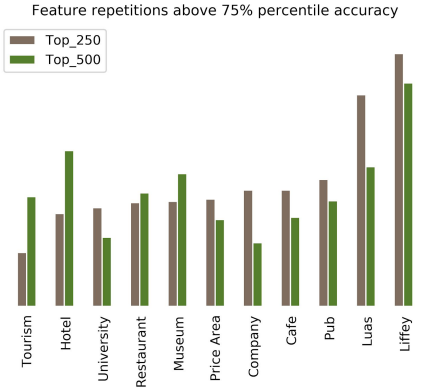
	combo	accuracy	accuracy_train	f1_score	jaccard_score	random_state	nFeatures
57759	['University', 'Cafe', 'Luas', 'Company']	0.734940	0.620061	0.734246	0.580177	84	4
57464	['University', 'Cafe', 'Luas']	0.710843	0.623100	0.710087	0.550594	84	3

Top_500

	combo	accuracy	accuracy_train	f1_score	jaccard_score	random_state	nFeatures
27839	['Hotel', 'Cafe', 'Pub', 'Luas', 'Liffey', 'Pr...	0.693252	0.628086	0.691741	0.529290	39	6
28265	['Hotel', 'University', 'Pub', 'Luas', 'Restau...	0.693252	0.634259	0.690474	0.528201	39	7



Let's keep those combos that pass the threshold of percentile 75% accuracy, and count the times each feature appears.



Technically we should have a look just at the combinations itself, not at individual features. However, this chart shows us that there are 2 predominant features for the Top_250 model ('Luas' & 'Liffey') and similarly 4 for Top_500 model ('Luas', Liffey', 'Hotel' & 'Museum').

b) SVM Cross Validation using K-Fold and cross_val_score

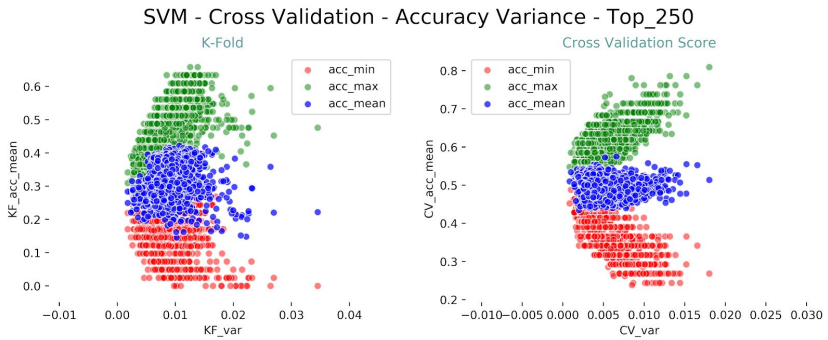
SVM K-Fold & Cross Validation Score for Top_250

K-Fold max accuracy

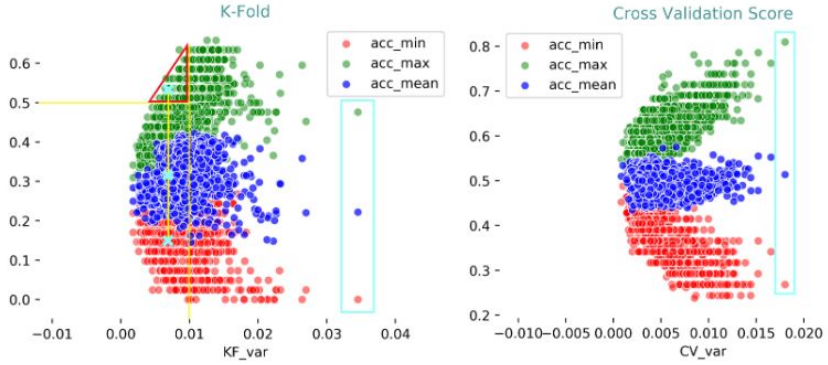
	combo	KF_acc_mean	KF_acc_max	KF_acc_min	KF_var	CV_acc_mean	CV_acc_max	CV_acc_min	CV_var	nFeatures
549	[Luas, Museum, Company, Liffey]	0.405168	0.658537	0.268293	0.0125537	0.514576	0.595238	0.428571	0.00251352	4
759	[Hotel, Luas, Museum, Company, Liffey]	0.395354	0.658537	0.268293	0.0138426	0.521893	0.609756	0.428571	0.00355541	5
1618	[Hotel, University, Cafe, Pub, Luas, Company, ...	0.388444	0.634146	0.243902	0.0119532	0.519396	0.609756	0.463415	0.00197534	7
1785	[University, Cafe, Pub, Luas, Restaurant, Comp...	0.383391	0.634146	0.219512	0.0142668	0.529036	0.619048	0.439024	0.00384321	7
1226	[Hotel, Cafe, Pub, Luas, Company, Liffey]	0.37619	0.634146	0.268293	0.0105816	0.519338	0.609756	0.487805	0.00168295	6

Cross_val_score max accuracy

	combo	KF_acc_mean	KF_acc_max	KF_acc_min	KF_var	CV_acc_mean	CV_acc_max	CV_acc_min	CV_var	nFeatures
128	[Tourism, Pub, Restaurant]	0.290592	0.500000	0.024390	0.014941	0.513821	0.809524	0.268293	0.018012	3.0
46	[Pub, Museum]	0.318118	0.463415	0.024390	0.018765	0.554936	0.785714	0.365854	0.015198	2.0
496	[Cafe, Pub, Museum, Restaurant]	0.276249	0.523810	0.024390	0.014193	0.537979	0.785714	0.365854	0.013164	4.0
474	[University, Luas, Museum, Price Area]	0.283391	0.439024	0.073171	0.011522	0.541405	0.780488	0.439024	0.008455	4.0
178	[Cafe, Pub, Company]	0.293844	0.439024	0.243902	0.003579	0.518815	0.761905	0.317073	0.012749	3.0



In order to find the best combination of features for the model we must get the group with the highest accuracy and lowest variance for K-Fold and Cross Validation and see if we get a match.
For example, high accuracy & low variance on the density chart would be the points on the left side of the green points. Each point on each color group represents one combination of features. In this case we might start looking for points with > 0.5 acc & < 0.01 variance on the KFold and see if we get a match on the Cross Validation side.

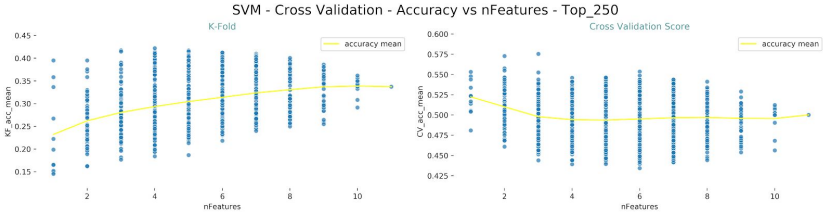


For each single combination of features, point in the chart in a color group, there are mirrors in the other 2 groups, due to these points sharing the same variance. A clear example is shown by the outliers on the right side (3 dots within the light blue box). These 3 points represent a unique combination of features. Following this thought, on the Cross Validation chart we notice a feature set with the highest accuracy, however it also has one of the worst. Therefore we are interested in the dots on the left hand side of each chart.

Let's define some conditions:

- K-Fold < 0.01 variance & > 0.5 accuracy
- CV < 0.005 variance & > 0.6 accuracy
- combo always above mean() for same nFeature

Applying these conditions we got 344 samples on the KFold group and 425 samples on the cross_val_score group.



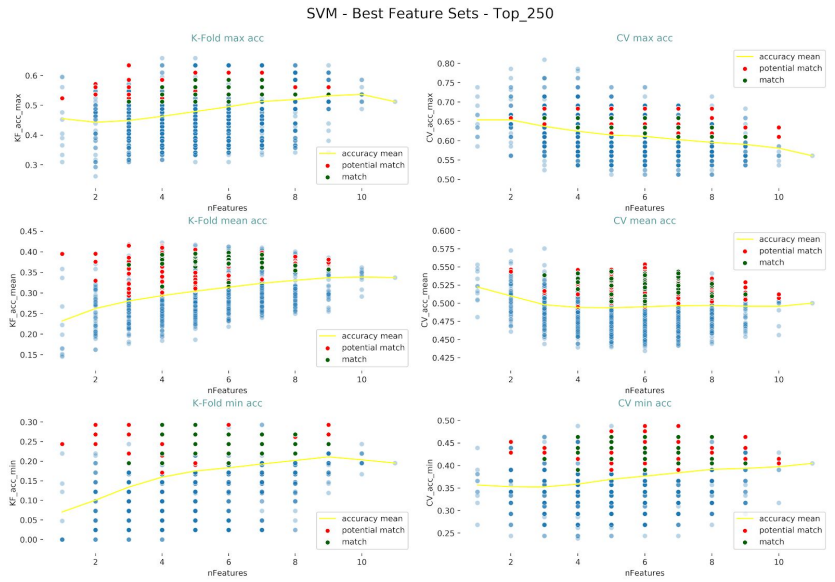
We first save the accuracy values of the grouped by nFeatures. These values represent the yellow accuracy mean on the previous chart

Now we create new dataframes applying the third condition:

- combo always above mean() for same nFeature

KFold group has been reduced from 344 to 216 samples
CV group has been reduced from 425 to 265 samples

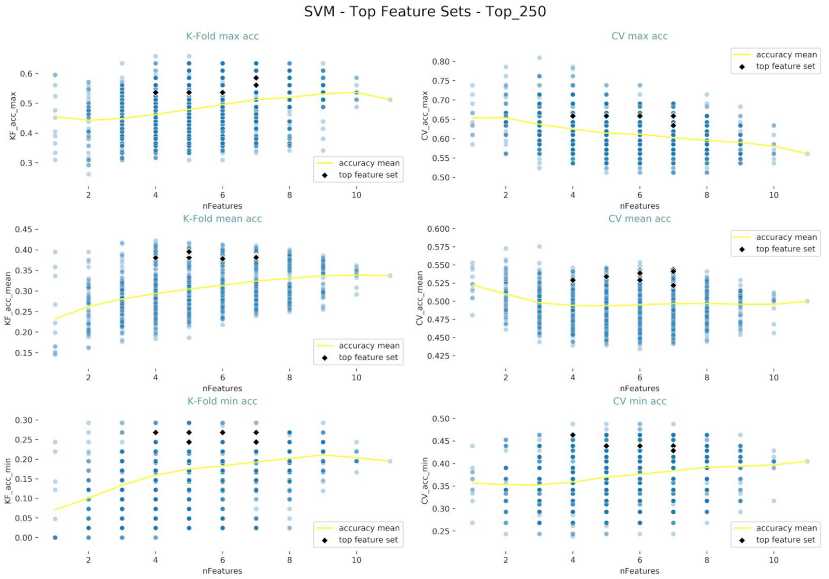
We compare KFold & CV groups to get matches and we got 65 matches



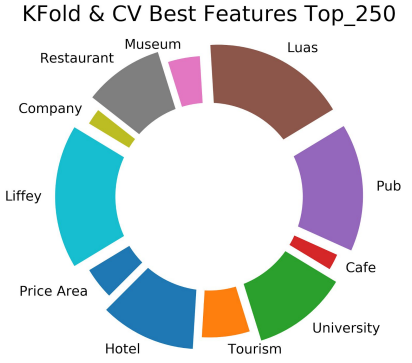
This is great, but we can get the Top 10 Feature Sets by increasing the values of the accuracy mean (yellow line) until we get 10 items in the match_list.

There are 9 best combos with delta 1.05. Delta means we had to increment the accuracy means by 5% until we obtained less than 10 matches.

	CV_acc_max	CV_acc_mean	CV_acc_min	CV_var	KF_acc_max	KF_acc_mean	KF_acc_min	KF_var	combo	nFeatures
25	0.658537	0.529152	0.463415	0.004023	0.536585	0.380778	0.268293	0.008047	[Hotel, Pub, Luas, Liffey]	4.0
65	0.658537	0.533972	0.439024	0.003769	0.536585	0.380952	0.268293	0.007386	[Hotel, University, Luas, Restaurant, Liffey]	5.0
85	0.658537	0.534030	0.439024	0.003833	0.536585	0.393031	0.268293	0.009040	[Tourism, Pub, Luas, Liffey, Price Area]	5.0
92	0.658537	0.533972	0.439024	0.004007	0.536585	0.395470	0.243902	0.008173	[University, Pub, Luas, Restaurant, Liffey]	5.0
124	0.658537	0.538792	0.439024	0.004303	0.536585	0.378571	0.268293	0.007556	[Hotel, University, Pub, Luas, Restaurant, Lif...	6.0
147	0.658537	0.529210	0.439024	0.004795	0.536585	0.378397	0.268293	0.009621	[Tourism, Pub, Luas, Restaurant, Liffey, Price...	6.0
164	0.634146	0.543612	0.439024	0.003940	0.560976	0.385714	0.243902	0.009375	[Hotel, Tourism, University, Cafe, Pub, Luas, ...	7.0
176	0.658537	0.541289	0.439024	0.003219	0.560976	0.383333	0.243902	0.009936	[Hotel, University, Pub, Luas, Museum, Restaur...	7.0
177	0.658537	0.521951	0.428571	0.003989	0.585366	0.381243	0.268293	0.009740	[Hotel, University, Pub, Luas, Museum, Company...	7.0

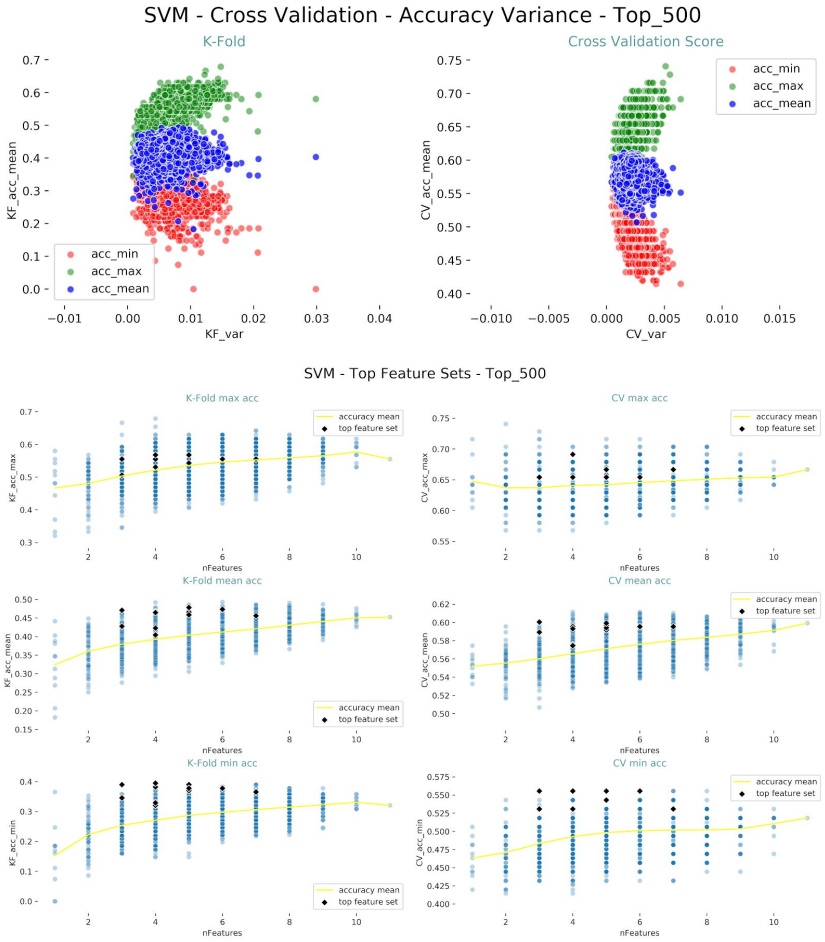


SVM Top_250 K-Folds accuracy: 38.42 %
SVM Top_250 Cross Validation accuracy: 53.4 %
In both cases the Top feature set are between 4 and 7 features



SVM K-Fold & Cross Validation Score for Top_500

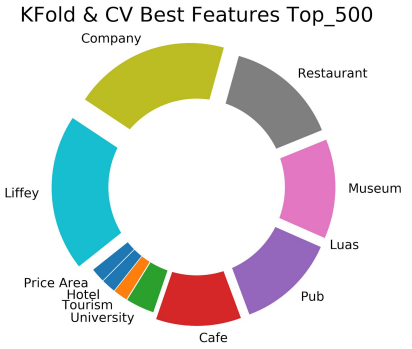
(following the same methodology that with Top_250)



Top_500 K-Folds accuracy: 45.45 %

Top_500 Cross Validation accuracy: 59.19 %

Top feature sets are between 3 and 7 features.



c) SVM KFold train set accuracy

(considering best combinations of feature set)

Top_250

n Feature set: 9

Number of results to calculate the average: 9 x 10 Folds = 90

SVM K-Fold Train accuracy mean for best feature sets Top_250: 63.81 %

Top_500

n Feature set: 12

Number of results to calculate the average: 12 x 10 Folds = 120

SVM K-Fold Train accuracy mean for best feature sets Top_500: 62.97 %

d) SVM ROC Curve

ROC displays a curve with a point for each of the True Positive Rate and False Positive Rate of the model at distant threshold levels, letting us see the compensation between the FPR and TPR for all threshold levels.

Top_250

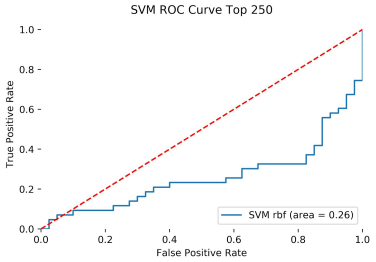
	combo	accuracy	accuracy_train	f1_score	jaccard_score	random_state	nFeatures
57759	['University', 'Cafe', 'Luas', 'Company']	0.734940	0.620061	0.734246	0.580177	84	4
57464	['University', 'Cafe', 'Luas']	0.710843	0.623100	0.710087	0.550594	84	3

acc: 0.7349397590361446

f1: 0.7342462874754834

jaccard: 0.5801769168175971

1st sample with the highest accuracy reports good accuracy, however with a high false positive rate.

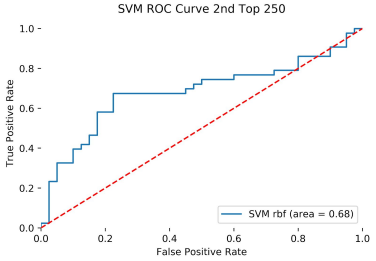


2nd sample

acc: 0.7108433734939759

f1: 0.7100868590641637

jaccard: 0.550593984328924



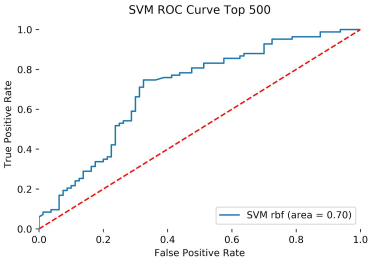
Top_500

	combo	accuracy	accuracy_train	f1_score	jaccard_score	random_state	nFeatures
27839	['Hotel', 'Cafe', 'Pub', 'Luas', 'Liffey', 'Pr...	0.693252	0.628086	0.691741	0.529290	39	6
28265	['Hotel', 'University', 'Pub', 'Luas', 'Restau...	0.693252	0.634259	0.690474	0.528201	39	7

acc: 0.6932515337423313

f1: 0.691741029931214

jaccard: 0.5292904066453119



e) SVM Validation

As we left 40 samples out of the scope to train the classifiers, let's see how they perform.

Notice that all samples are negative, this means that in order to get 100% accuracy the classifier should predict 40 negatives and zero positives.

Top_250

Method	Predicted Negative	Predicted Positive	Accuracy %
random_state	25	15	62.5
KFold	21	19	52.5
cross_val_score	18	22	45.0

Top_500

Method	Predicted Negative	Predicted Positive	Accuracy %
random_state	23	17	57.5
KFold	20	20	50.0
cross_val_score	26	14	65.0

2. K-Nearest Neighbors Classifier

KNN has been used because the dataset is small enough, and it is one of the most used learning algorithms. KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point

- We are running the KNN classifier for all combinations of features.
- random_state from 0 to 100 and multiple of 3. This will serve as Cross Validation. Otherwise, fixing the seed (random_state) which generates the test set, wouldn't get optimal results.
- Data has been splitted in 80/20 ratio, 80% for train set and 20% for test set.

a) KNN Cross Validation iterating random state parameter

KNN accuracy improved without median Zero and target values (0, 1)

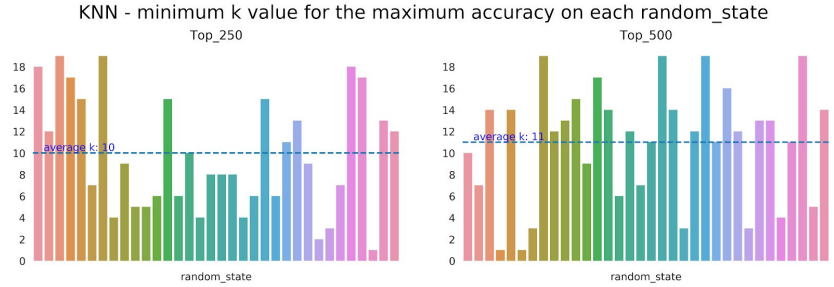
Top_250

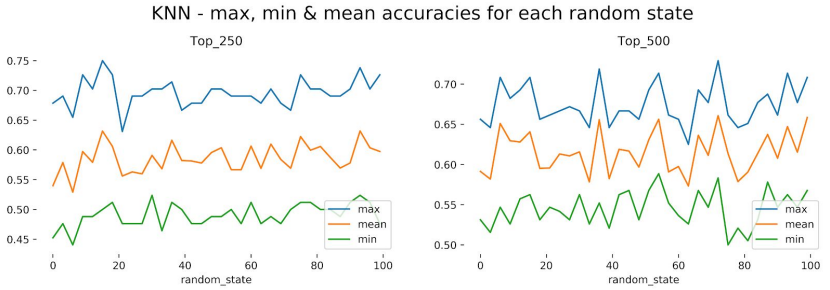
	combo	accuracy	k	f1_score	jaccard_score	random_state	nFeatures
519	[Cafe, Museum, Company, Liffey]	0.75	10.0	0.565267	0.395804	15.0	4.0
387	[Tourism, Cafe, Luas, Company]	0.75	7.0	0.580420	0.409722	15.0	4.0

Top_500

	combo	accuracy	k	f1_score	jaccard_score	random_state	nFeatures
735	[Hotel, Cafe, Company, Liffey, Price Area]	0.729167	16.0	0.639183	0.481204	72.0	5.0
18	[Hotel, Company]	0.723958	18.0	0.694792	0.537506	72.0	2.0

Next chart is showing for each random_state, the minimum k value for the maximum accuracy calculated for all 2047 feature set combinations





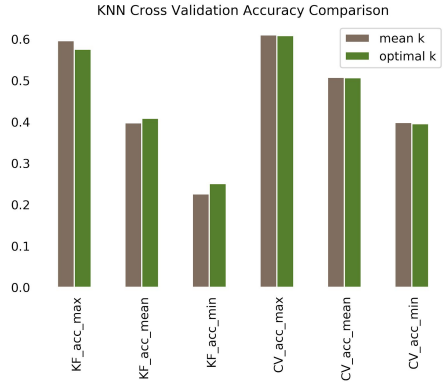
b) KNN Cross Validation using K-Fold and cross_val_score

As we have estimated, the average k value (neighbours within range) is:

- 10 for Top_250
- 11 for Top_500

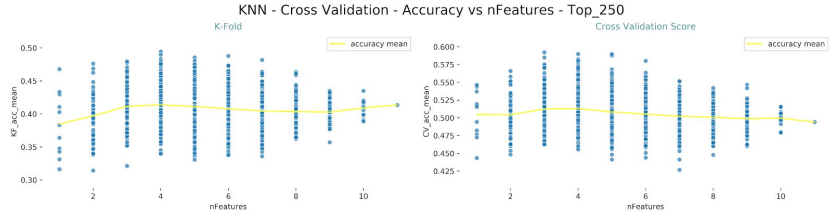
However, if we use the average k value with our script which iterates for all possible combinations of features(combos), we would not get the optimal results. Fortunately, we already had calculated the best k value for different random_state. So we can calculate the average k value for each combo using 34 random_state per combo.

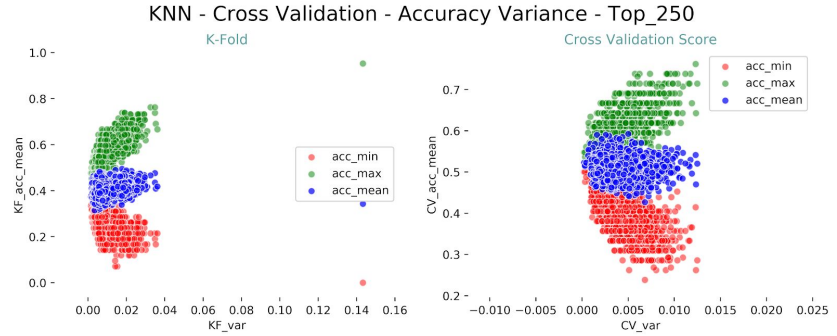
Then we just include this k value within the combo's loop.



KNN K-Fold & Cross Validation Score Top_250

	combo	KF_acc_mean	KF_acc_max	KF_acc_min	KF_var	CV_acc_mean	CV_acc_max	CV_acc_min	CV_var	nFeatures	knn_k
10	[Price Area]	0.342973	0.952381	0	0.143265	0.515447	0.595238	0.428571	0.002476	1	6
167	[University, Museum, Liffey]	0.434901	0.761905	0.190476	0.0348706	0.525087	0.585366	0.439024	0.00167652	3	6

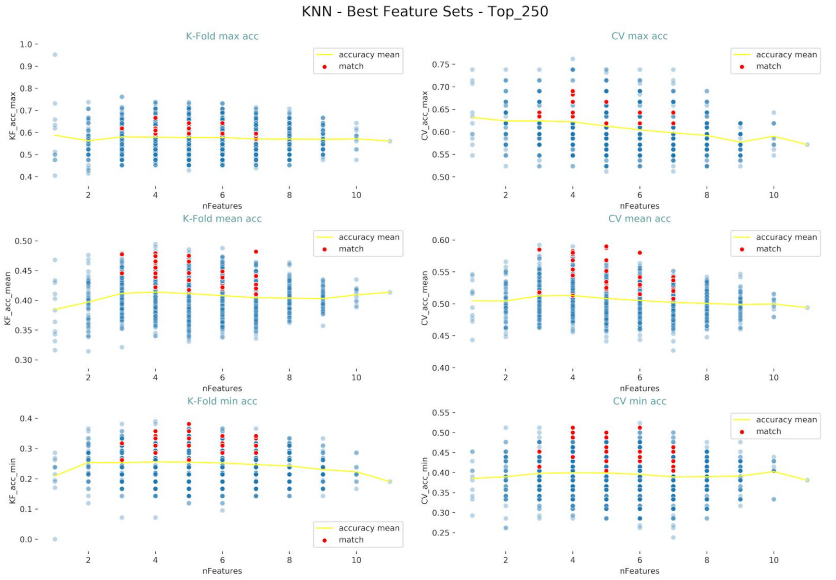




We can spot an outlier on the right hand side of the K-Fold chart.
Now we can apply the same methodology that with the SVM classifier. First at all, let's define some initial conditions in order to find the best combos (feature set):

- K-Fold < 0.01 variance & > 0.5 accuracy
- CV < 0.005 variance & > 0.6 accuracy
- combo always above mean() for same nFeature

We got 30 matches

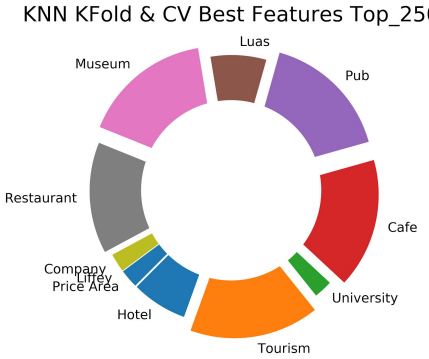


We have 30 different feature sets for our model. We can narrow the group, so let's get the Top 10 Feature Sets as we did with the SVM classifier. We just start increasing the accuracy mean (yellow line) until we get 10 items in the match_list. There are 9 best combos with delta 1.03 (We increased 3% the mean in order to obtain 9 matches)

	CV_acc_max	CV_acc_mean	CV_acc_min	CV_var	KF_acc_max	KF_acc_mean	KF_acc_min	KF_var	combo	knn_k	nFeatures
22	0.666667	0.568002	0.487805	0.002007	0.609756	0.463008	0.357143	0.006719	[Hotel, Pub, Luas, Restaurant]	5.0	4.0
28	0.666667	0.577642	0.487805	0.003234	0.666667	0.479152	0.341463	0.007244	[Tourism, Cafe, Pub, Museum]	13.0	4.0
29	0.690476	0.582520	0.487805	0.003739	0.619048	0.465679	0.309524	0.009320	[Tourism, Cafe, Museum, Restaurant]	12.0	4.0
30	0.690476	0.580139	0.500000	0.004047	0.619048	0.474448	0.341463	0.007451	[Tourism, Pub, Museum, Restaurant]	9.0	4.0
32	0.642857	0.553717	0.500000	0.003060	0.609756	0.465563	0.333333	0.007678	[Cafe, Pub, Luas, Museum]	8.0	4.0
38	0.666667	0.587573	0.500000	0.002923	0.619048	0.474739	0.357143	0.005449	[Hotel, Tourism, Cafe, Pub, Museum]	11.0	5.0
56	0.666667	0.589895	0.500000	0.002569	0.642857	0.475029	0.333333	0.006859	[Tourism, Cafe, Pub, Museum, Restaurant]	12.0	5.0
69	0.642857	0.580139	0.512195	0.002131	0.642857	0.448722	0.285714	0.009552	[Hotel, Tourism, Cafe, Pub, Museum, Restaurant]	10.0	6.0
107	0.619048	0.520093	0.428571	0.003467	0.595238	0.419861	0.309524	0.006337	[Tourism, University, Cafe, Luas, Restaurant, ...	7.0	7.0

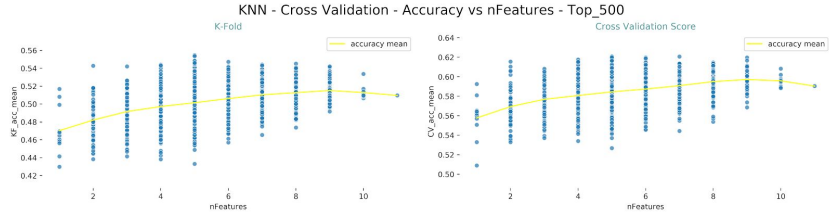
Top_250 K-Folds accuracy: 46.29 %
Top_250 Cross Validation accuracy: 57.11 %

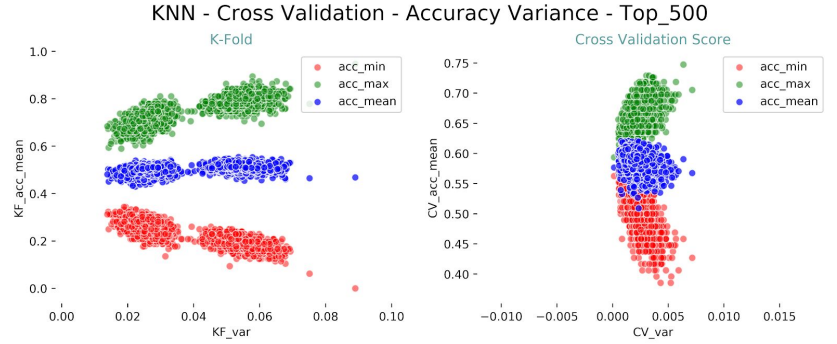
Top feature sets are between 4 and 7 features



KNN K-Fold & Cross Validation Score Top_500
(following same methodology which we used with Top_250)

	combo	KF_acc_mean	KF_acc_max	KF_acc_min	KF_var	CV_acc_mean	CV_acc_max	CV_acc_min	CV_var	nFeatures	knn_k
10	[Price Area]	0.468202	0.947368	0	0.0889921	0.508925	0.589474	0.427083	0.00233279	1	6
1021	[Luas, Restaurant, Company, Liffey, Price Area]	0.553607	0.894737	0.229167	0.0578291	0.616579	0.684211	0.53125	0.0021251	5	12

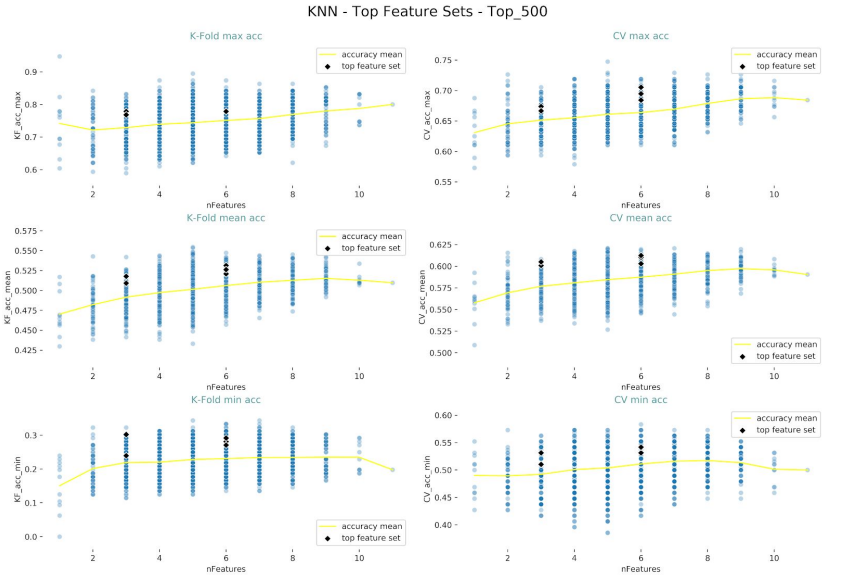




Same code that earlier with different initial values (we pick them just having a look on the chart). We don't have to be accurate on this, due to we will be filtering results until we get the optimal feature set for our model using the KNN classifier and Cross Validation in this case.

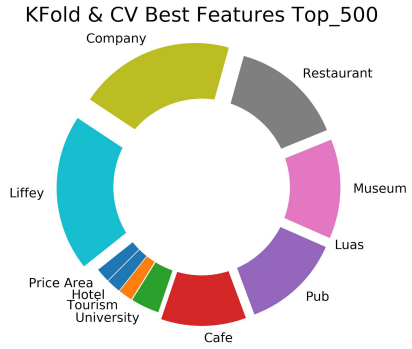
KFold group has been reduced from 912 to 61 samples
CV group has been reduced from 1701 to 444 samples
We got 31 matches
There are 5 best combos with delta 1.02

	CV_acc_max	CV_acc_mean	CV_acc_min	CV_var	KF_acc_max	KF_acc_mean	KF_acc_min	KF_var	combo	knn_k	nFeatures
4	0.673684	0.600921	0.531250	0.001715	0.778947	0.509276	0.302083	0.024181	[Cafe, Liffey, Price Area]	11.0	3.0
6	0.666667	0.605110	0.510417	0.002249	0.768421	0.517818	0.239583	0.028180	[Luas, Restaurant, Liffey]	11.0	3.0
34	0.694737	0.610362	0.541667	0.002003	0.778947	0.531382	0.281250	0.024440	[Hotel, Pub, Luas, Museum, Liffey, Price Area]	11.0	6.0
38	0.684211	0.603037	0.531250	0.002099	0.778947	0.520943	0.270833	0.027230	[Tourism, Cafe, Luas, Company, Liffey, Price A...	11.0	6.0
44	0.705263	0.612412	0.541667	0.001809	0.778947	0.526162	0.291667	0.026363	[Luas, Museum, Restaurant, Company, Liffey, Pr...	11.0	6.0



Top_500 K-Folds accuracy: 52.11 %
Top_500 Cross Validation accuracy: 60.64 %

Top feature sets are between 3 and 6 features



c) KNN KFold train set accuracy

(considering best combinations of feature set)

Top_250

n Feature set: 9

Number of results to calculate the average: 9 x 10 Folds = 90

KNN K-Fold Train accuracy mean for best feature sets Top_250: **66.28 %**

Top_500

n Feature set: 5

Number of results to calculate the average: 5 x 10 Folds = 50

KNN K-Fold Train accuracy mean for best feature sets Top_500: **69.04 %**

d) KNN ROC Curve

ROC displays a curve with a point for each of the True Positive Rate and False Positive Rate of the model at distant threshold levels, letting us see the compensation between the FPR and TPR for all threshold levels.

Top_250

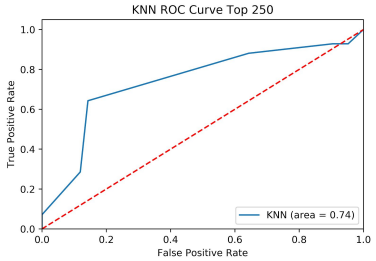
	combo	accuracy	k	f1_score	jaccard_score	random_state	nFeatures
10754	['Cafe', 'Museum', 'Company', 'Liffey']	0.75	10	0.565267	0.395804	15	4
10622	['Tourism', 'Cafe', 'Luas', 'Company']	0.75	7	0.580420	0.409722	15	4

acc: 0.75

f1: 0.7470967741935484

jaccard: 0.5970394736842105

We got much worse accuracy if we use the stratify parameter while splitting the data.



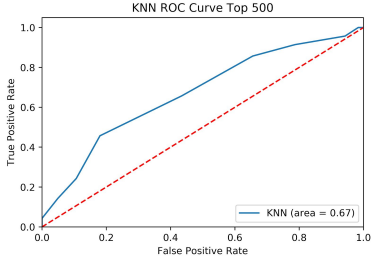
Top_500

	combo	accuracy	k	f1_score	jaccard_score	random_state	nFeatures
49863	['Hotel', 'Cafe', 'Company', 'Liffey', 'Price ...	0.729167	16	0.639183	0.481204	72	5
49146	['Hotel', 'Company']	0.723958	18	0.694792	0.537506	72	2

acc: 0.6875

f1: 0.676954094292804

jaccard: 0.5239470108695652



e) KNN Validation

As we left 40 samples out of the scope to train, let's see how they perform.
Notice that all samples are negative, this means that in order to get 100% accuracy the classifier should predict 40 negatives and zero positives.

Top_250

Method	Predicted Negative	Predicted Positive	Accuracy %
random_state	130	97	57.27
KFold	120	107	52.86
cross_val_score	116	111	51.10

Top_500

Method	Predicted Negative	Predicted Positive	Accuracy %
random_state	186	41	81.94
KFold	185	42	81.50
cross_val_score	183	44	80.62

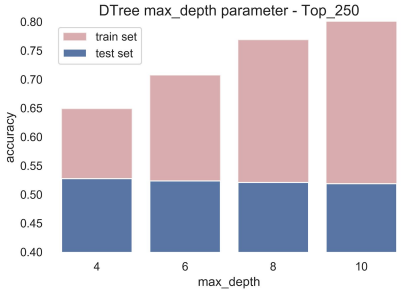
3. Decision Tree Classifier

Decision tree is one of the fastest way to identify most significant variables and the relationship between two or more variables. Albeit it is useful during the data exploration stage, in this project would be interesting to see the feature sets we obtain with DTree and compare them with SVM and KNN

a) Cross Validation iterating random state parameter

After some testing the input of a dataset with mean zero (gaussian distribution) has improved results. Same with SVM. Binomial target as (1,0) or (1,-1) does not affect results at all.

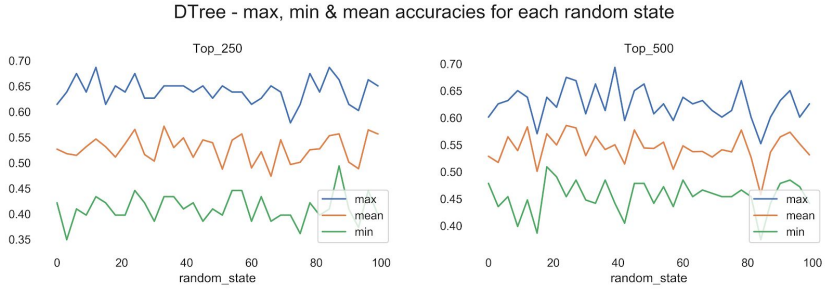
DTree accuracy improved with median Zero and target values (-1, 1)



depth_max set the maximum splits a tree can make before giving a prediction. In this case, with many repetitions of this process we have forced it to an extremely classification tree with many nodes overfitting the training set. Therefore, we are using just the data with 4 depths.

Test accuracy mean Top_250: 52.76 %
Train accuracy mean Top_250: 64.91 %
Test accuracy mean Top_500: 54.4 %
Train accuracy mean Top_500: 62.68 %

These results are for all combinations, not for the optimal feature set, which will be calculated using KFold & cross_validation.



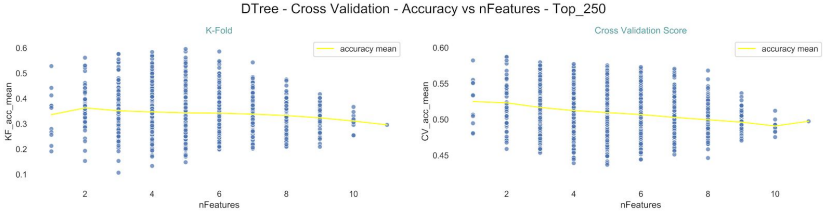
b) DTree Cross Validation using K-Fold and cross_val_score

Top_250

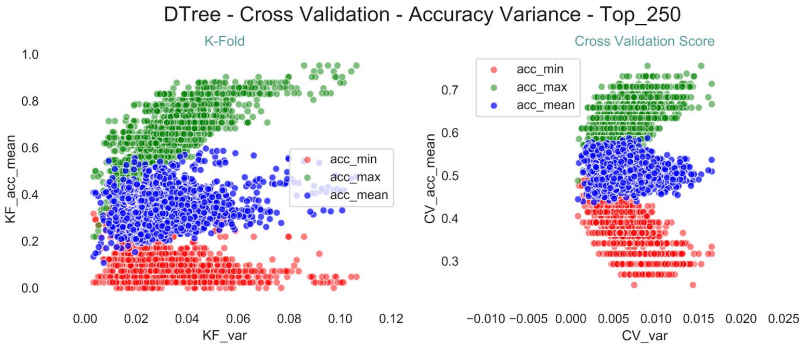
	combo	KF_acc_mean	KF_acc_max	KF_acc_min	KF_var	CV_acc_mean	CV_acc_max	CV_acc_min	CV_var	nFeatures
657	[Hotel, University, Cafe, Museum, Company]	0.542451	0.95122	0.219512	0.0857699	0.526481	0.642857	0.439024	0.00312423	5
1035	[Hotel, Tourism, University, Cafe, Museum, Com...	0.535075	0.95122	0.146341	0.0917929	0.50482	0.609756	0.390244	0.00336481	6

Top_500

	combo	KF_acc_mean	KF_acc_max	KF_acc_min	KF_var	CV_acc_mean	CV_acc_max	CV_acc_min	CV_var	nFeatures
1390	[Tourism, Pub, Museum, Restaurant, Company, Li...	0.458913	0.814815	0.271605	0.0272769	0.553689	0.654321	0.506173	0.0027756	6
1483	[Pub, Museum, Restaurant, Company, Liffey, Pri...	0.434222	0.814815	0.271605	0.021529	0.549985	0.654321	0.481481	0.00291571	6



DTree K-Fold & Cross Validation Score Top_250



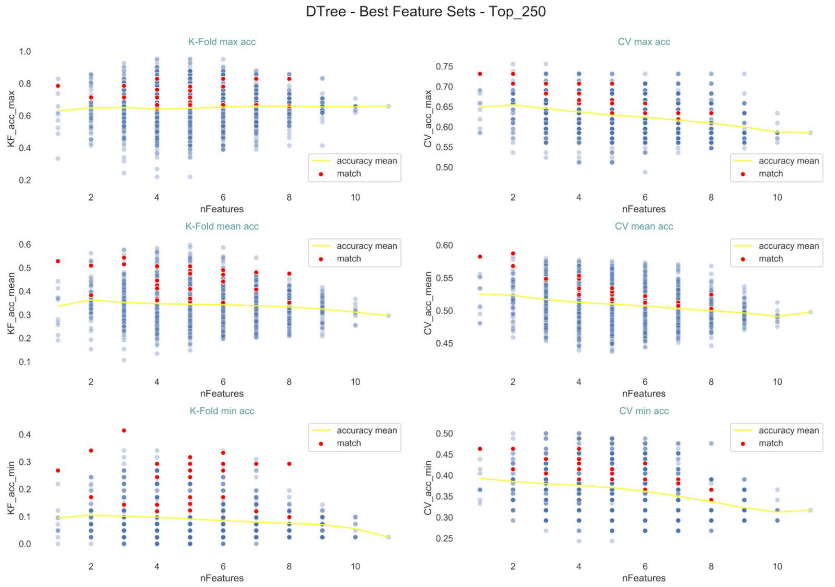
First thing we notice is that the highest accuracy values have also the highest variance (max & min accuracies within a feature set more dispersed)

Now we can apply the same methodology with SVM classifier. First at all, let's define some initial conditions in order to find the best combos (feature set):

- K-Fold < 0.03 variance & > 0.5 accuracy
- CV < 0.007 variance & > 0.6 accuracy
- combo always above mean() for same nFeature

KFold group has been reduced from 733 to 100 samples
CV group has been reduced from 798 to 315 samples

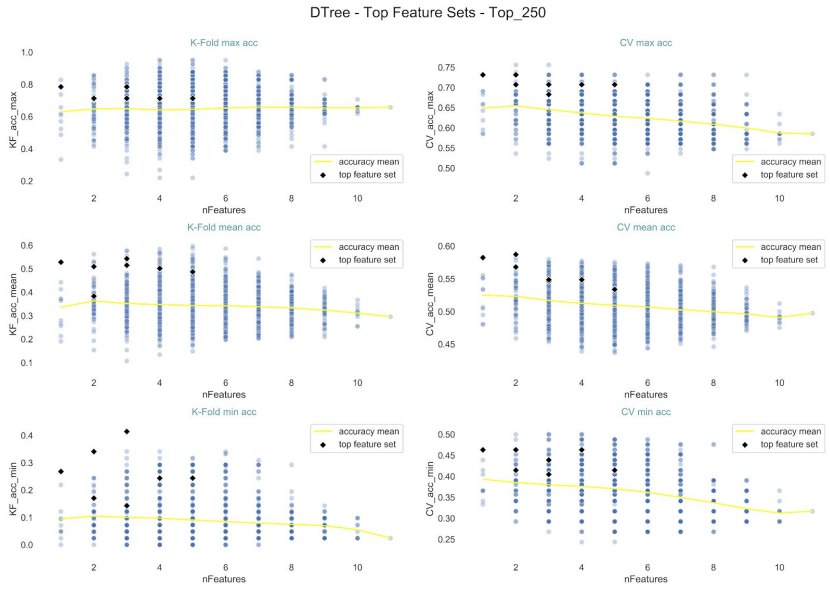
We compare KFold & CV groups to get matches
We got 28 matches



We have 28 different feature sets for our model. We can narrow the group, so let's get the Top 10 Feature Sets. We just start increasing the accuracy mean (yellow line) until we get 10 items in the match_list

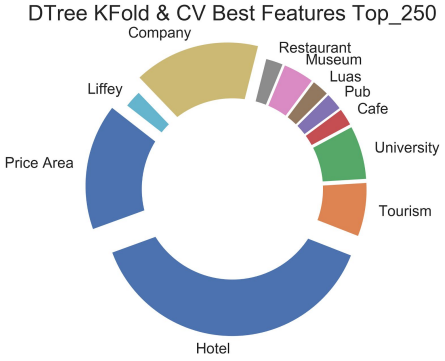
There are 7 best combos with delta 1.05

	CV_acc_max	CV_acc_mean	CV_acc_min	CV_var	KF_acc_max	KF_acc_mean	KF_acc_min	KF_var	combo	nFeatures
0	0.731707	0.582346	0.463415	0.006048	0.785714	0.528397	0.268293	0.020955	[Hotel]	1.0
2	0.731707	0.587166	0.414634	0.006806	0.707317	0.383391	0.170732	0.027815	[Hotel, Museum]	2.0
3	0.707317	0.567944	0.463415	0.006755	0.714286	0.509233	0.341463	0.011526	[Hotel, Company]	2.0
7	0.682927	0.546283	0.404762	0.006964	0.785714	0.514808	0.142857	0.029979	[Hotel, Tourism, Price Area]	3.0
11	0.707317	0.548606	0.439024	0.005524	0.714286	0.543148	0.414634	0.009110	[Hotel, Company, Price Area]	3.0
22	0.707317	0.548548	0.463415	0.005570	0.714286	0.501626	0.243902	0.015763	[Hotel, University, Company, Price Area]	4.0
42	0.707317	0.533856	0.414634	0.006848	0.714286	0.486992	0.243902	0.024281	[Hotel, Tourism, University, Company, Price Area]	5.0

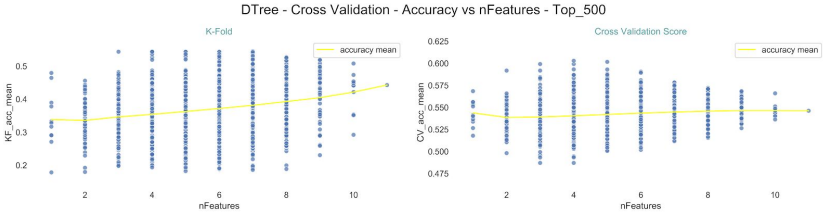


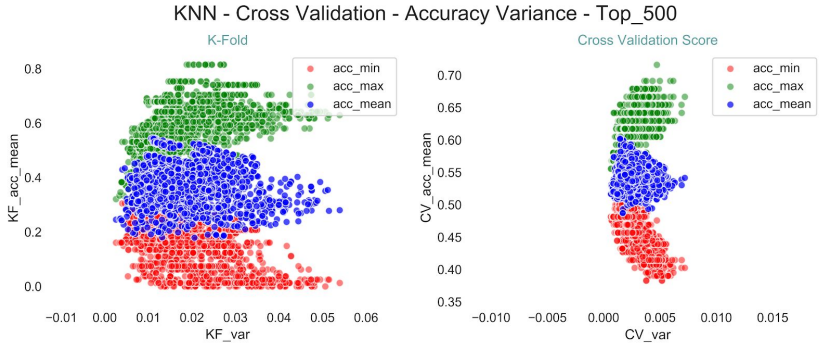
DTree Top_250 K-Folds accuracy: 49.54 %
DTree Top_250 Cross Validation accuracy: 55.93 %

Top feature sets for Top_250 are between 1 and 5 features



DTree K-Fold & Cross Validation Score Top_500
(following the same methodology that with Top_250)

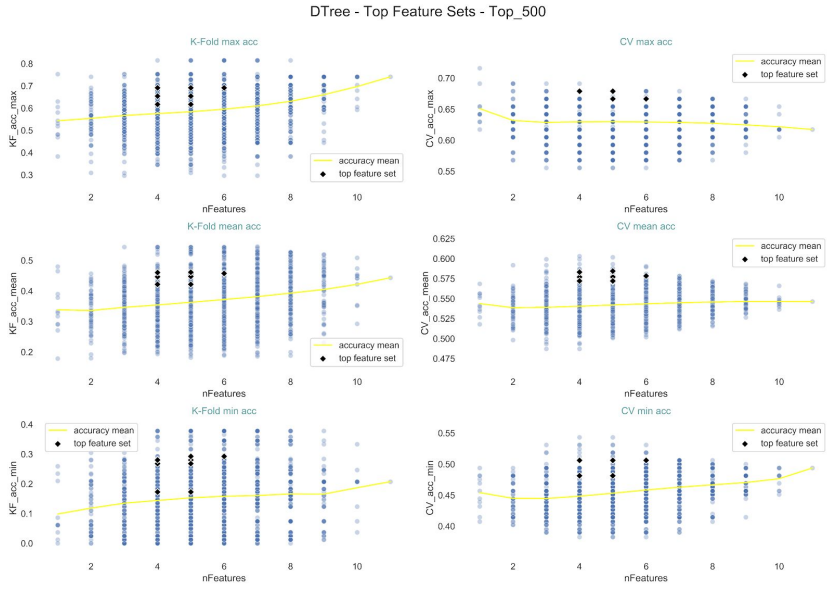




Same code that earlier with different initial values (we pick them just having a look on the chart. We don't have to be accurate on this, due to we'll filter the results later till we get the optimal feature set for out model using the DTree classifier and Cross Validation in this case

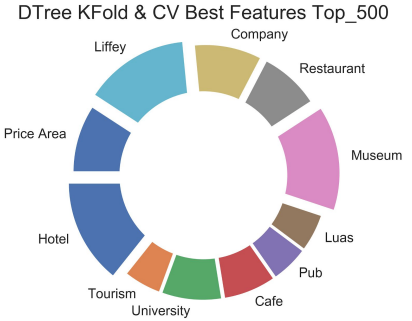
KFold group has been reduced from 885 to 346 samples
CV group has been reduced from 1815 to 448 samples
We got 116 matches
There are 9 best combos with delta 1.06

	CV_acc_max	CV_acc_mean	CV_acc_min	CV_var	KF_acc_max	KF_acc_mean	KF_acc_min	KF_var	combo	nFeatures
57	0.679012	0.577085	0.506173	0.002603	0.654321	0.421816	0.172840	0.019363	[Hotel, Cafe, Museum, Liffey]	4.0
69	0.679012	0.572162	0.481481	0.002777	0.617284	0.446582	0.268293	0.010994	[Hotel, Museum, Restaurant, Liffey]	4.0
70	0.679012	0.583273	0.506173	0.002605	0.691358	0.460148	0.280488	0.014665	[Hotel, Museum, Company, Liffey]	4.0
144	0.666667	0.570927	0.481481	0.002283	0.617284	0.451521	0.268293	0.011832	[Hotel, University, Museum, Restaurant, Liffey]	5.0
145	0.666667	0.578335	0.506173	0.002366	0.691358	0.456444	0.280488	0.015210	[Hotel, University, Museum, Company, Liffey]	5.0
154	0.679012	0.577085	0.506173	0.002603	0.654321	0.421816	0.172840	0.019363	[Hotel, Cafe, Museum, Liffey, Price Area]	5.0
166	0.679012	0.572162	0.481481	0.002777	0.617284	0.447817	0.268293	0.011246	[Hotel, Museum, Restaurant, Liffey, Price Area]	5.0
167	0.679012	0.584508	0.506173	0.002520	0.691358	0.461367	0.292683	0.014241	[Hotel, Museum, Company, Liffey, Price Area]	5.0
246	0.666667	0.578335	0.506173	0.002366	0.691358	0.457663	0.292683	0.014794	[Hotel, University, Museum, Company, Liffey, P...	6.0



DTree Top_500 K-Folds accuracy: 44.72 %
DTree Top_500 Cross Validation accuracy: 57.71 %

Top feature sets are between 4 and 6 features



c) DTree KFold train set accuracy

(considering best combinations of feature set)

Top_250
n Feature set: 7
Number of results to calculate the average: 7 x 10 Folds = 70
DTree K-Fold Train accuracy mean for best feature sets Top_250: **64.12 %**

Top_500
n Feature set: 9
Number of results to calculate the average: 9 x 10 Folds = 90
DTree K-Fold Train accuracy mean for best feature sets Top_500: **65.3 %**

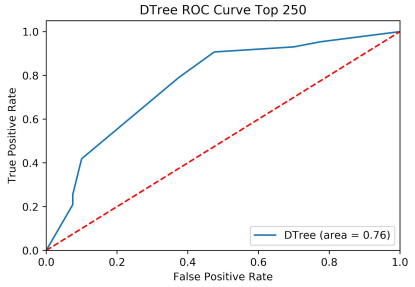
d) DTree ROC Curve

ROC displays a curve with a point for each of the True Positive Rate and False Positive Rate of the model at distant threshold levels, letting us see the compensation between the FPR and TPR for all threshold levels.

Top_250

	combo	accuracy	accuracy_train	max_depth	f1_score	jaccard_score	random_state	nFeatures
147564	['Cafe', 'Pub', 'Price Area']	0.722892	0.650456	8	0.711495	0.555895	12	3
128749	['Hotel', 'Tourism', 'University', 'Cafe', 'Lu...	0.722892	0.702128	6	0.722892	0.566038	84	8

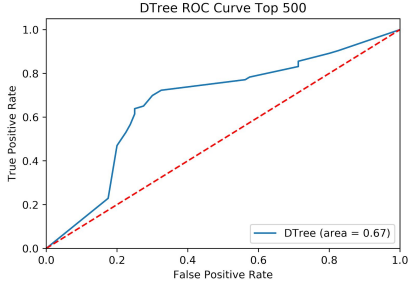
acc: 0.7228915662650602
f1: 0.7114948751594344
jaccard: 0.5558951347913649



Top_500

	combo	accuracy	accuracy_train	max_depth	f1_score	jaccard_score	random_state	nFeatures
156185	['Hotel', 'Tourism', 'Pub', 'Luas', 'Liffey']	0.699387	0.766975	8	0.699341	0.537732	24	5
27746	['Hotel', 'Tourism', 'Luas', 'Museum', 'Restau...	0.693252	0.649691	4	0.686749	0.524987	39	6

acc: 0.6993865030674846
f1: 0.6994091353653904
jaccard: 0.5377749746765285



e) DTree Validation

As we left 40 samples out of the scope to train the classifiers, let's see how they perform.
Notice that all samples are negative, this means that in order to get 100% accuracy the classifier should predict 40 negatives and zero positives.

Top_250

Method	Predicted Negative	Predicted Positive	Accuracy %
random_state	9	31	22.5
KFold	7	33	17.5
cross_val_score	27	13	67.5

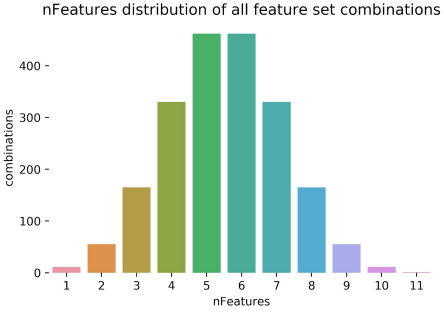
Top_500

Method	Predicted Negative	Predicted Positive	Accuracy %
random_state	24	16	60.0
KFold	29	11	72.5
cross_val_score	21	19	52.5

Diamond shape of charts versus nFeatures

For all three classifiers: SVM, KNN and DTree.

At a higher number of combinations (5 & 6 nFeatures) there is more probability to find extreme max and min (farest from the mean accuracy) running the classifier.



IV. RESULTS

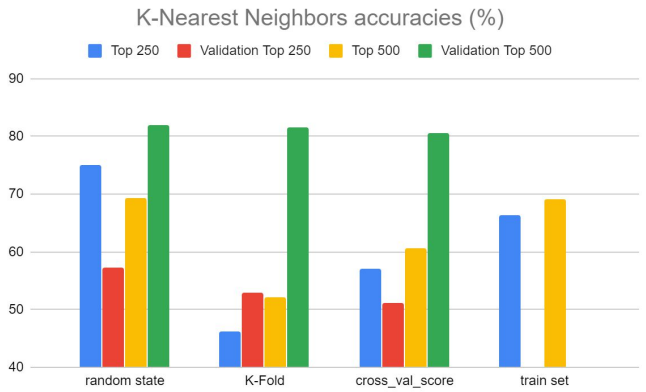
As we can see with the results, there are some points with noting:

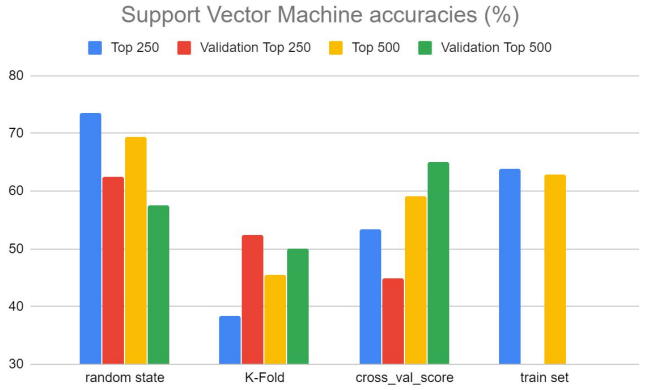
- KNN with more consistent results, especially once the classifier was trained with the entire dataset, predicting the validation set with an accuracy above 80%.

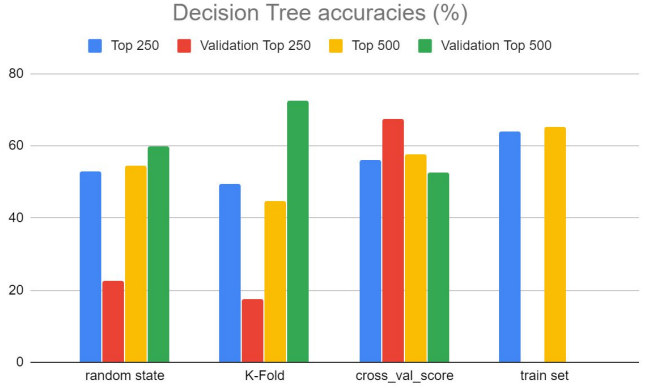
- My approach of applying Cross Validation using the random state parameter with SVM and KNN, has obtained clearly better results than using instead the K-Fold or cross_val_score classifiers from Scikit. While for the DTree classifier the results were similar.

- Regarding the use of two different targets in parallel, we noticed consistent accuracy results between them. However, with the random state method the Top 250 has higher accuracies, while for the Top 500 the use of K-Fold and cross_val_score would be more appropriate.

- Train set was about 60% for the three approaches used.

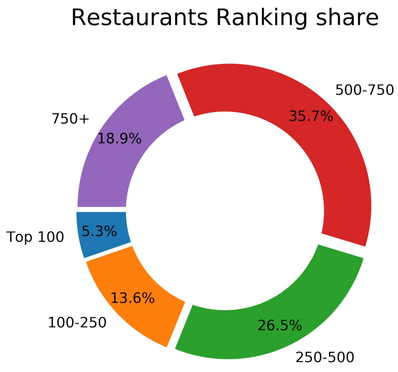






V. DISCUSSION

Really good accuracy, we fall into the accuracy Paradox though.



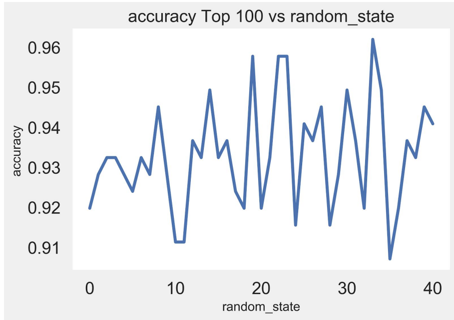
cumulative shares:

Top 100	5.3%
Top 250	18.9%
Top 500	45.4%
Top 750	81.1

I trained the model for the combinations that the entire feature set had. 11 features gives you 2047 combinations. I iterated the SVM for different train and test sets. As seeds are generated using the random_state parameter, I trained the model modifying this parameter within the range 0 to 40 for Cross Validation purposes. This generated a dataframe of 83927 samples, just for the Top_100.
I got an average 94% accuracy, which it might sound great, albeit reality was a bit different. All samples with the same random_state had the same results, independently to the number of features or even which ones were, as shown on the table below.

	combo	accuracy	f1_score	jaccard_score	random_state	nFeatures
83926	(Hotel, Tourism, University, Cafe, Pub, Luas, ...	0.940928	0.912291	0.885346	40	11
82903	(Hotel, Tourism, University, Cafe, Pub, Luas)	0.940928	0.912291	0.885346	40	6
82553	(Hotel, University, Pub, Museum, Liffey)	0.940928	0.912291	0.885346	40	5
82554	(Hotel, University, Pub, Museum, Price Area)	0.940928	0.912291	0.885346	40	5
82555	(Hotel, University, Pub, Restaurant, Company)	0.940928	0.912291	0.885346	40	5
82556	(Hotel, University, Pub, Restaurant, Liffey)	0.940928	0.912291	0.885346	40	5
82557	(Hotel, University, Pub, Restaurant, Price Area)	0.940928	0.912291	0.885346	40	5
82558	(Hotel, University, Pub, Company, Liffey)	0.940928	0.912291	0.885346	40	5
82559	(Hotel, University, Pub, Company, Price Area)	0.940928	0.912291	0.885346	40	5
82560	(Hotel, University, Pub, Liffey, Price Area)	0.940928	0.912291	0.885346	40	5

To train the model, I gave values equal to 1 if the sample was within Top 100, and 0 if it was not. *y-variance* (Top_100) was 0.06, which means the model was clearly



imbalanced. The explanation of the singular results is simple: if I have a group where 95% of samples are False training the classifier, we will get in return a 95% accuracy predicting always that a new sample will be False.

This is why the Prediction test set *array ŷ* was always 0.

```
yhat
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

To solve this we need to understand how the SVM classifier works better.
From scikit-learn.org: "*Standardization of datasets is a common requirement for many machine learning estimators implemented in scikit-learn; they might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with zero mean and unit variance.*"

Moreover, the values that we give to the target (Top_100) matter. We need a standard deviation (variance) of 1, which means we need to assign negatives values to them. Doing this we got a *y-variance* of 0.34 (before was 0.06), although the model still imbalanced enough to take the results of Top_100 valid.

F1-measure, which weights precision and recall equally, is the variant most often used when learning from imbalanced data. However, even though obtaining high F1 values I took the decision on not using Top_100 in the model. As a minimum, *y-variance* of 0.50 or greater must be required to train the SVM classifier.

VI. CONCLUSION

In this project I have tried to balance the amount of text and graphics. In some chapters I have focused on the use of charts in order to demonstrate clearly, in a stepwise manner, the approach which I have taken to explain and rationalise my thinking on this project to the reader.

I believe my decision in using an approach with all 2047 combinations of features, has given the project not just an overview of the accuracy of results which can be obtained from the use of this model, but also, and more importantly, a clear insight into how the different algorithms, inherent in the model, can be used to identify and recognise correlations between the features which are the subject of this project.

Dublin has a nuclear distribution around the city center which has increased the difficulty in using classifiers as a means of demonstrating the capability of the model. As a follow on project, I believe it would be interesting to investigate this approach in a city with a different social infrastructure, for example, a city built within the last two centuries. The city of Dublin can trace its origin back over a thousand years.

For instance, large metropolitan cities in North America and evolving “smart cities” in South Korea and China would potentially provide an alternative outcome and further insight into the capability of the model.

Even though, repeating the calculations for the Top 250 and Top 500 models increased the overall time spent, I was curious to see the final results for both, and which model would potentially be a better indicator for future projects. Although we did not obtain a clear winner, the results have been interesting, showing good results on particular methods for the Top 250, while Top 500 has been under performing. Perhaps this is due to the nature and complexity of the data, which as I mentioned before, the centralised infrastructure of the city is having an important influence on how the algorithms behave.

VII. REFERENCES

https://www.dublinchamber.ie/business-agenda/economic-profile-of-dublin
https://www.bordbia.ie/globalassets/bordbia2020/industry/news/2019_irish_foodservice_marketconsumer_insights_final.pdf
https://www.luas.ie/
https://www.daft.ie/
https://www.tripadvisor.ie/
https://www.idaireland.com/doing-business-here
https://stackoverflow.com/
https://scikit-learn.org/
https://www.dublinlive.ie/lifestyle/travel
https://www.openstreetmap.org/

https://towardsdatascience.com/anova-for-feature-selection-in-machine-learning-d9305e228476
https://towardsdatascience.com/the-5-feature-selection-algorithms-every-data-scientist-need-to-know-3a6b566efd2
https://towardsdatascience.com/project-report-for-data-science-coding-exercise-9a9c76a09be8
https://towardsdatascience.com/svm-and-kernel-svm-fed02bef1200

https://machinelearningmastery.com/evaluate-performance-machine-learning-algorithms-python-using-resampling/
https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/
https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/

http://cs229.stanford.edu/proj2014/Kyle%20Carbon,%20Kacyn%20Fujii,%20Prasanth%20Verina,%20Applications%20Of%20Machine%20Learning%20To%20Predict%20Yelp%20Ratings.pdf
Hotel location evaluation: A combination of machine learning tools and web GIS
https://www.researchgate.net/publication/331346839_Machine_learning_applications_to_smart_city
https://www.sciencedirect.com/science/article/abs/pii/S0278431915000316?via%3Dihub
https://academic.oup.com/bioinformatics/article/21/8/1509/249540
Statistical Feature Selection

VIII. ACKNOWLEDGMENT

I would like to thank the Python Community, especially the *Stackoverflow community*, where many of my questions had been solved in the past, so I was able to learn and improve my coding skills during this project.

I would also like to extend my thanks to all open source and free resources which have been used in this project, such as Scikit learn and Anaconda.

Albeit goals define your path,
the best part still being the journey.

IX. APPENDIX

The code used for this project can be found at:
https://github.com/jotagar

For a better user experience, I recommend using Jupyter nbviewer to access to the repository:
https://nbviewer.jupyter.org/github/jotagar/